REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	31 March 1998	**Technical Report**

4. TITLE AND SUBTITLE
Proof Polynomials: A Unified Semantics for Modality and λ–terms

5. FUNDING NUMBERS
DAAH04-96-1-0341

6. AUTHOR(S)
S. N. Artemov

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Regents of the University of California
c/o Sponsored Projects Office
336 Sproul Hall
Berkeley, CA 94720-5940

Cornell Univ.
625 Rhodes Hall
Ithaca, NY 14853

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 35873.7-MA-MUR

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by the documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
In this paper we present a finite axiomatization of the Logic of Proofs in Hilbert, Gentzen and natural deductions systems and prove normalization theorms.

19980519 117

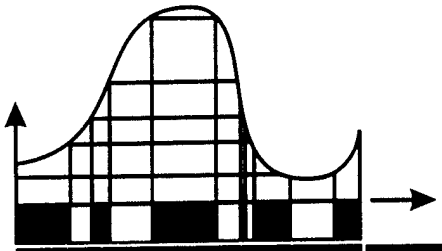
14. SUBJECT TERMS
logic of proofs, intuitionistic logic, proof polynomials, λ–terms

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

DTIC QUALITY INSPECTED 2



Center for Foundations of Intelligent Systems

Technical Report
98-06

Proof Polynomials: A Unified Semantics for Modality and λ-terms

S. N. Artemov

March 1998

CORNELL
UNIVERSITY

625 Rhodes Hall, Ithaca, NY 14853 (607) 255-8005

Technical Report
98-06

Proof Polynomials: A Unified Semantics for Modality and λ-terms

S. N. Artemov

March 1998

Proof polynomials: a unified semantics for modality and λ-terms

Sergei N. Artemov *

March, 1998

Abstract

In 1933 Gödel ([16]) introduced a modal logic of provability ($\mathcal{S}4$) but left it without an intended mathematical model. In [17] Gödel suggested the format *"t is a proof of F"* ($t:F$) to understand $\mathcal{S}4$. This suggestion is realized in the *Logic of Proofs* ($\mathcal{LP}$) presented in this paper. In $\mathcal{LP}$, proofs are internalized as terms. Three operators *application* ($\cdot$), *choice* ($+$), and *proof checker* ($!$) are used to construct terms called *proof polynomials* from atomic terms. $\mathcal{LP}$ enables iteration of the internalization of proofs as terms thus allowing formulas $t:F$ for any proof polynomial t and any formula F.

In this paper we present a finite axiomatization of $\mathcal{LP}$ in Hilbert, Gentzen and natural deduction systems and prove normalization theorems. Three completeness properties of $\mathcal{LP}$ are established: i) proof polynomials represent all operations on proofs invariant with respect to the proof system chosen; ii) $\mathcal{LP}$ is sound and complete with respect to the provability semantics; iii) $\mathcal{LP}$ is able to realize any derivation in $\mathcal{S}4$ by recovering proof polynomials for all occurrences of modality in a given $\mathcal{S}4$-derivation. Thus $\mathcal{LP}$ gives an intended semantics for the Gödel's logic of provability $\mathcal{S}4$.

Logic of Proofs is the underlying structure for intuitionistic logic, some modal, epistemic logics, and typed lambda calculi. In particular, the basic $\mathcal{LP}$ propositions $t:F$ also can be interpreted as "*λ-term t of type F*". Following Curry-Howard we represent a λ-term $t:F$ with free variables $\{x_1:A_1,\ldots,x_n:A_n\}$ by a natural deduction in $\mathcal{LP}$ of the form $x_1:A_1,\ldots,x_n:A_n \Rightarrow t:F$. $\mathcal{LP}$ includes the λ-calculi corresponding to intuitionistic and modal logics. In addition $\mathcal{LP}$ admits λ-terms of the form $p:(t:F)$ and in general allows arbitrary combinations of ":" and propositional connectives. $\mathcal{LP}$ can also be seen as a typing system for non-deterministic λ-calculi since $t:F$ and $t:G$ are allowed in $\mathcal{LP}$ for different F and G. Proof polynomials provide a unified semantics for modalities and λ-terms. Such a unified semantics allows us to consider both modalities and λ-terms as objects of the same nature.

*Center for Foundations of Intelligent Systems, Rhodes Hall 625, Cornell University, Ithaca NY, 14853 email:artemov@hybrid.cornell.edu. Research supported by the ARO under the MURI program "Integrated Approach to Intelligent Systems", grant number DAA H04-96-1-0341.

1 Introduction to Logic of Proofs

In 1932 Kolmogorov ([20]) suggested to understand (propositional) intuitionistic logic $\mathcal{I}nt$ as a *calculus of problems* in classical mathematics. Kolmogorov gave a detailed but informal description of his interpretation. In 1933 Gödel ([16]) derived $\mathcal{I}nt$ from the classical notion of *proof* which may be regarded a special case of Kolmogorov's *problem solution.* Gödel reduced $\mathcal{I}nt$ to the classical modal *logic of provability* $\mathcal{S}4$, which has as axioms all classical tautologies in the modal propositional language along with the principles $\Box F \to F$, $\Box(F \to G) \to (\Box F \to \Box G)$, $\Box F \to \Box\Box F$, and rules *modus ponens* $F \to G, F \vdash G$ and *necessitation* $F \vdash \Box F$. However, Gödel did not give a precise provability model for $\mathcal{S}4$ itself. Kleene realizability [19], Medvedev finite problems [29] (cf. [44], [12]) provide readings of $\mathcal{I}nt$ as a logic of classical problems, but neither of these models specify $\mathcal{I}nt$ exactly.

Another problem arose from Gödel's paper [16]: to find an interpretation of provability, which complies with $\mathcal{S}4$. The issue of a provability model for $\mathcal{S}4$ attracted the attention of Kripke [22], Montague [34], Myhill [35],[36], Lemmon [25], Mints [31], Kuznetsov [24], Goldblatt [18], Boolos [8],[9] Shapiro [39],[39], Buss [11], and many other logicians. None of the semantics suggested serve as a faithful provability model for $\mathcal{S}4$. Some of them do not treat the $\mathcal{S}4$ modality as a provability operator (e.g. [22]), some capture only a fragment of $\mathcal{S}4$ (e.g. [25]). Semantics from [24], [18], [8], [9] capture some non-$\mathcal{S}4$ formulas (e.g. so-called Grzegorcsyk scheme) that are not plausible as the principles of the Provability.

A principal difficulty here is caused by the reflexivity principle $\Box F \to F$ of $\mathcal{S}4$, which along with the necessitation rule $F \vdash \Box F$ contradicts the Second Gödel Incompleteness Theorem (*SGIT*). Indeed, a straightforward interpretation of $\Box F$ as

"*F is provable in a formal system* $\mathcal{T}$"

leads to logics of formal provability incompatible with $\mathcal{S}4$ (cf.[8],[9]). For example, a *reflexivity principle* $\Box F \to F$ under an interpretation $F = A \wedge \neg A$) and $\Box$ as a provability within $\mathcal{T}$ becomes a statement *Consis* $\mathcal{T}$, expressing the consistency of $\mathcal{T}$. A theorem of $\mathcal{S}4$ $\Box(\Box F \to F)$ thus expresses an assertion that *Consis* $\mathcal{T}$ is provable in $\mathcal{T}$, which contradicts *SGIT*.

The following analysis explains this paradoxical behavior. The arithmetical formula of provability *Provable*(F) for *F is provable in* $\mathcal{T}$ has the form

"*there exists a number x which is the code of a proof of F*".

In a given model of first order arithmetic such an x may be nonstandard. In particular, it would mean that *Provable*(F) is true, but there is no "real" $\mathcal{T}$-derivation behind such an x. So, *Provable*(F) is weaker than the intended "*F is provable*". Hence

$$\mathit{Provable}(F) \to F$$

is stronger, than the intended

"*if F is provable, then F*".

This consideration suggests eliminating the existential quantifiers and replacing them by explicit operations on proofs. Where should these operations come from? The proof of *SGIT* says more about Provability than $\Box$ is able to express. For example, the usual distributivity formula

$$\Box(F \to G) \to (\Box F \to \Box G)$$

is a "forgetful" version of

> *"given proofs s and t of $F \to G$ and F respectively $m(s,t)$ is a proof of G, where $m(x,y)$ is a specific computable function built from the description of a proof system"*

A similar decoding can be done for $\Box F \to \Box\Box F$:

> *"for some specific function $c(x)$ given a proof t of F $c(t)$ returns a proof that t is a proof of F"* .

In his Lecture at Zilsel's ([17], cf. [37])) in 1938 (published in 1995) Gödel sketched a constructive version of $\mathcal{S}4$ in which *t is a proof of F* ($t : F$) replaced the $\mathcal{S}4$ assertion *F is provable* ($\Box F$). Gödel's system of 1938 justifies the reflexivity principle, its propositional part admits an exact provability interpretation. However, the question of finding a complete set of axioms for a logic of proofs, as well as the question about its ability to realize the entire $\mathcal{S}4$ remained unanswered.

In this paper we realize Gödel's suggestion of 1938[1] and find a complete system for the logic of proofs. This yields a positive solution to the problem of faithful provability semantics for Gödel's provability logic of 1933, which in turn proves Kolmogorov's conjecture of 1932 that intuitionistic logic $\mathcal{I}nt$ is nothing but a logic of proofs ("problem solutions") for systems based on classical logic.

1.1 Definition. The language of Logic of Proofs ($\mathcal{LP}$) contains

the usual language of classical propositional logic
proof variables $x_0, \ldots, x_n, \ldots$, proof constants $a_0, \ldots, a_n, \ldots$
functional symbols: monadic !, binary $\cdot$ and $+$
operator symbol of the type "*term : formula*".

We will use $a, b, c, \ldots$ for proof constants, and $u, v, w, x, y, z, \ldots$ for proof variables. Terms are defined by the grammar

$$p ::= x_i \mid a_i \mid !p \mid p_1 \cdot p_2 \mid p_1 + p_2$$

[1]The Logic of Proofs $\mathcal{LP}$ was found by the author independently of Gödel's paper [17]. The first presentations of $\mathcal{LP}$ took place at the author's talks at the conferences in Münster and Amsterdam in 1994. Preliminary versions of some of the results of this paper appeared in Technical Reports [4] and [5].

We call these terms *proof polynomials* and denote them by $p,r,s,t\ldots$. By analogy we refer to constants as coefficients. Constants correspond to proofs of a finite fixed set of propositional schemas. We will also omit $\cdot$ whenever it is safe. We also assume that $(a \cdot b \cdot c)$, $(a \cdot b \cdot c \cdot d)$, *etc.* should be read as $((a \cdot b) \cdot c)$, $(((a \cdot b) \cdot c) \cdot d)$, *etc.*

Using t to stand for any term and S for any propositional letter, the formulas are defined by the grammar

$$\sigma ::= S \mid \sigma_1 \to \sigma_2 \mid \sigma_1 \wedge \sigma_2 \mid \sigma_1 \vee \sigma_2 \mid \neg\sigma \mid t{:}\sigma$$

We will use $A, B, C, F, G, H, X, Y, Z$ for the formulas in this language, and $\Gamma, \Delta, \ldots$ for the finite sets (also finite multisets, or finite lists) of formulas unless otherwise explicitly stated. We will also use $\vec{x}, \vec{y}, \vec{z}, \ldots$ and $\vec{p}, \vec{r}, \vec{s}, \ldots$ for vectors of proof variables and proof polynomials respectively. If $\vec{s} = \{s_1, \ldots, s_n\}$ and $\Gamma = \{F_1, \ldots, F_n\}$, then $\vec{s}{:}\Gamma$ denotes $\{s_1{:}F_1, \ldots, s_n{:}F_n\}$, $\bigvee\Gamma = F_1 \vee \ldots \vee F_n$, $\bigwedge\Gamma = F_1 \wedge \ldots \wedge F_n$.

We assume the following precedences from highest to lowest:

$$!, \cdot, +, :, \neg, \wedge, \vee, \to .$$

The intended semantics for $p{:}F$ is

p is a proof of F,

which will be formalized in the next section. Note that a proof p is not necessarily deterministic, i.e. p may be a proof of several different F's. The differences between deterministic and nondeterministic proofs are mostly cosmetic. The same proof may be easily considered as deterministic, e.g. as a proof of the end formula of the proof tree, and as nondeterministic, say as a proof of all intermediate formulas derived within this proof. In an abstract setting every nondeterministic proof system can be made deterministic by changing from

$$\text{“}p \text{ proves } F_1, \ldots, F_n\text{”} \qquad \text{to} \qquad \text{“}(p,i) \text{ proves } F_i,\ i = 1, \ldots n\text{”}.$$

Moreover, every deterministic proof system may be regarded as nondeterministic by reading

$$\text{“}p \text{ proves } F_1 \wedge \ldots \wedge F_n\text{”} \qquad \text{as} \qquad \text{“}p \text{ proves each of } F_i,\ i = 1, \ldots n\text{”}.$$

The logic of strictly deterministic proof systems was discussed in [2], [3], [23], where it meets a complete axiomatization. Such logic is considerably more complicated than $\mathcal{LP}$. It lacks some natural closure properties and it is not able to realize any modal logic, in particular $\mathcal{S}4$. The situation here resembles that with the definition of the computable functions. Total computable functions (as well as deterministic proof systems) may look more desirable than general computable functions (as well as general proof systems without discrimination on the basis of determinisity). However, a separate theory of total computable functions lacks the elegance and some important closure properties of the theory of general computability.

The system $\mathcal{LP}_0$. Axioms:

A0. Axioms of classical propositional logic in the language of $\mathcal{LP}$

A1. $t{:}F \to F$ *"verification"*

A2. $t{:}(F \to G) \to (s{:}F \to (t{\cdot}s){:}G)$ *"application"*

A3. $t{:}F \to !t{:}(t{:}F)$ *"proof checker"*

A4. $s{:}F \to (s{+}t){:}F, \quad t{:}F \to (s{+}t){:}F$ *"choice"*

Rule of inference:

R1. $$\frac{\Gamma \vdash F \to G \qquad \Gamma \vdash F}{\Gamma \vdash G}$$ *"modus ponens".*

The system $\mathcal{LP}$ is $\mathcal{LP}_0$ plus the rule

R2. if $\mathbf{A}$ *is an axiom A0 - A4, and c a proof constant, then* $\dfrac{}{\vdash c{:}\mathbf{A}}$ *"necessitation"*

Proof constants in $\mathcal{LP}$ stand for proofs of "simple facts", namely propositional axioms and axioms *A1 - A4*. We define a *Constant Specification (CS)* to be a finite set of formulas $c_1 : A_1, \ldots, c_n : A_n$ such that c_i is a constant, and F_i an axiom *A0 - A4*. Each derivation in $\mathcal{LP}$ naturally generates a $\mathcal{CS}$ consisting of all formulas introduced in this derivation by the *necessitation* rule.

Note that both $\mathcal{LP}_0$ and $\mathcal{LP}$ enjoy the deduction theorem and the substitution lemma: *If* $\Gamma(x, P) \vdash B(x, P)$ *for a propositional variable* P *and a proof variable* x*, then for any proof polynomial* t *and any formula* F

$$\Gamma(x/t, P/F) \vdash B(x/t, P/F).$$

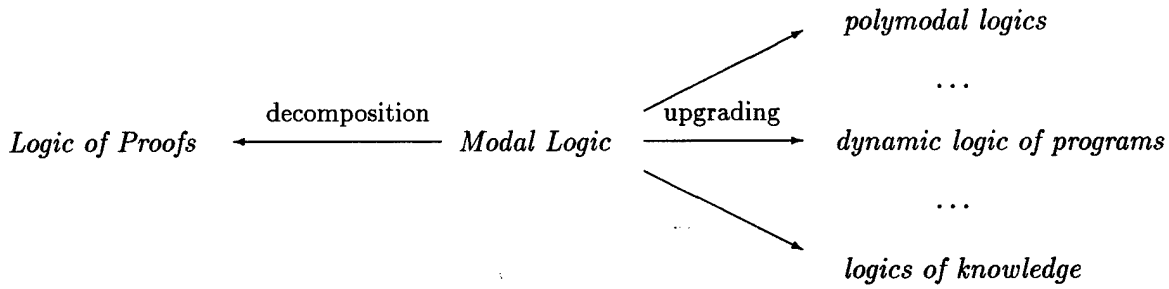
For a given constant specification $\mathcal{CS}$ under $\mathcal{LP}(\mathcal{CS})$ we mean $\mathcal{LP}_0$ plus CS. Obviously,

F is derivable in $\mathcal{LP}$ with a constant specification $\mathcal{CS}$ $\Leftrightarrow$ $\mathcal{LP}(\mathcal{CS}) \vdash F$ $\Leftrightarrow$ $\mathcal{LP}_0 \vdash \bigwedge \mathcal{CS} \to F$.

Operations "$\cdot$" and "!" may work on deterministic as well as on non-deterministic proofs. In turn, "+" is a strictly non-deterministic operation. Indeed, by *A4* we have $s : F \wedge t : G \to (s{+}t){:}F \wedge (s{+}t){:}G$, thus $s + t$ proves different formulas. In principle, there is no restriction on the choice of a constant c in *R2*. In particular, R2 allows to introduce a formula $c{:}A(c)$, or to specify a constant several times as a proof of different axioms from *A0 - A4*. One might restrict $\mathcal{LP}$ to injective constant specifications only, i.e. only allowing each constant to serve as a proof of a single axiom $\mathbf{A}$ within a given derivation (although allowing constructions $c{:}\mathbf{A}(c)$, as before). Such a restriction would not change the ability of $\mathcal{LP}$ to emulate classical modal logic, or the functional and arithmetical completeness theorems for $\mathcal{LP}$, though it will provoke an excessive renaming of the constants. We choose not to put the injectivity restriction on

$\mathcal{LP}$: derivations with injective $\mathcal{CS}$'s lack some natural closure properties, e.g. it might be the case that $\vdash F$ and $\vdash G$ but not $\vdash F \wedge G$. Since proof polynomials denote nondeterministic proofs it seems natural that the proof constants should be non-deterministic too. One cannot *a priori* assign a constant to a given axiom from $A1$ - $A4$ for all derivations in $\mathcal{LP}$: such derivations are not closed under substitution.

No single operator "$t:$" in $\mathcal{LP}$ is a normal modality since none of them satisfies the property $t:(P \to Q) \to (t:P \to t:Q)$. The usual Kripke semantics for modal logics does not work for the Logic of Proofs. These make $\mathcal{LP}$ fundamentally different from numerous multi modal logics, e.g. the dynamic logic of programs ([21]), where the modality is upgraded by some additional features. In turn, in the Logic of Proofs the modality is **decomposed** into a family of proof polynomials: $\mathcal{S}4$ is a forgetful projection of $\mathcal{LP}$ (see Section 6)



Gödel's sketch of the logic of proofs from [17] lacks "+", without which a realization of $\mathcal{S}4$ cannot be completed.

Gabbay's Labelled Deductive Systems (LDS) may also be regarded as a natural framework for the Logic of Proofs [14]. One of the original motivations of LDS was to elaborate a proof interpretation of atoms $t:F$. In the area of epistemic logics a desire to have a knowledge supporting system with justifications in the format $t:F$ was expressed in [6]. The Logic of Proofs may be regarded as a basic epistemic logic with explicit justifications. Intuitionistic Type Theory by Martin-Löf [27], [28] also makes use of the format $t:F$ with its informal provability reading.

Terms of typed combinatory logic (cf. [45]) may be regarded as proof polynomials in a fragment of $\mathcal{LP}$ with $\to$ and $\cdot$ only. To some extend $\mathcal{LP}$ as it is presented in this paper may be regarded as a combinatory logic system with nondeterministic assignments of terms to types and capacities to internalize its own derivations as combinatory terms. Of course, the solid provability semantics remains the main distinction of $\mathcal{LP}$. In principle one might rewrite $\mathcal{LP}$ as a λ-term system without proof constants but with a corresponding set of atomic λ-terms. In a term calculi language such rewriting would mean a transition from a combinatory

term system to a λ-term system of the same strength. From the logic point of view such rewriting would mean a change from Hilbert style proofs as a source for proof polynomials into equivalent natural deduction ones with a bunch of new rules instead. In this paper we choose the language of proof polynomials which looks equally close to (or equally distant from) the two major applications of $\mathcal{LP}$: modal logic and λ-calculus.

Among the related works there are [23], [41] and [33] all based on the system $\mathcal{LP}$ as it was presented in [4]. The paper [23] gives a complete axiomatization of the logic of deterministic proof systems (i.e. the ones where any proof proves exactly one theorem). The paper [41] finds a complete axiomatization of a joint logic of proofs and formal provability. In [33] the decidability of the entire $\mathcal{LP}$ was first established.

1.2 Lemma. (Lifting Lemma) *Let*

$$\vec{s}{:}\Gamma, \Delta \vdash_{\mathcal{LP}} F,$$

and let $\mathcal{D}$ be a corresponding derivation. Then one can construct a proof polynomial $t(\vec{x}, \vec{y})$ such that

$$\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash_{\mathcal{LP}} t(\vec{s}, \vec{y}){:}F.$$

Moreover, if

$$\Delta \vdash_{\mathcal{LP}} F,$$

then one can construct $t(\vec{y})$ which is a product of proof constants and variables from $\vec{y}$ such that

$$\vec{y}{:}\Delta \vdash_{\mathcal{LP}} t(\vec{y}){:}F.$$

Proof. By induction on the derivation $\vec{s}{:}\Gamma, \Delta \vdash F$. If $F = s_i{:}G_i \in \vec{s}{:}\Gamma$, then put $t := !s_i$ and use *A3*. If $F = D_j \in \Delta$, then put $t := y_j$. If F is an axiom *A0* - *A4*, then pick a fresh proof constant c and put $t := c$; by *R2*, $F \vdash c{:}F$. Let F be introduced by *modus ponens* from $G \to F$ and G. Then, by the induction hypothesis, there are proof polynomials $u(\vec{s}, \vec{y})$ and $v(\vec{s}, \vec{y})$ such that $u{:}(G \to F)$ and $v{:}G$ are both derivable in $\mathcal{LP}$ from $\vec{s}{:}\Gamma, \vec{y}{:}\Delta$. By *A1*, $\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash (uv){:}F$, and we put $t := uv$. If F is introduced by *R2*, then $F = c{:}A$ for some axiom A. Use the same *R2* followed by *A3*: $c{:}A \to !c{:}c{:}A$, to get $\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash !c{:}F$, and put $t := !c$.

◀

It is easy to see from the proof that the lifting polynomial $t(\vec{s}, \vec{y})$ is nothing but a blueprint of a given derivation $\vec{s}{:}\Gamma, \Delta \vdash F$. Thus $\mathcal{LP}$ accommodates its own proofs as proof terms. The necessitation rule

$$\vdash F \;\Rightarrow\; \vdash p{:}F \;\; \textit{for some proof polynomial } p,$$

where p is a blueprint of the proof $\vdash F$ is a special case of Lifting. As we can see in Section 6 $\mathcal{LP}$ suffices to emulate all derivations of $\mathcal{S}4$.

1.3 Example. $\mathcal{S}4 \vdash (\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$

In $\mathcal{LP}$ the corresponding derivation is

1. $A, B \vdash A \wedge B$, by propositional logic
2. $x{:}A, y{:}B \vdash t(x,y){:}(A \wedge B)$, by Lifting
3. $\vdash x{:}A \wedge y{:}B \rightarrow t(x,y){:}(A \wedge B)$, from 2.

1.4 Example. $\mathcal{S}4 \vdash (\Box A \vee \Box B) \rightarrow \Box(A \vee B)$.

In $\mathcal{LP}$ the corresponding derivation is

1. $A \rightarrow A \vee B, \quad B \rightarrow A \vee B$
2. $a{:}(A \rightarrow A \vee B), \quad b{:}(B \rightarrow A \vee B)$, by *necessitation*,
3. $x{:}A \rightarrow (a \cdot x){:}(A \vee B)$, from 2 by $A2$
4. $y{:}B \rightarrow (b \cdot y){:}(A \vee B)$, from 2 by $A2$
5. $ax{:}(A \vee B) \rightarrow (ax{+}by){:}(A \vee B)$,
   $by{:}(A \vee B) \rightarrow (ax{+}by){:}(A \vee B)$, by $A4$
6. $(x{:}A \vee y{:}B) \rightarrow (ax{+}by){:}(A \vee B)$

2 Standard provability interpretation of $\mathcal{LP}$

The Logic of Proofs is meant to play for a notion of proof a role similar to that played by the boolean propositional logic for the notion of statement. In particular, $\mathcal{LP}$ enjoys the soundness/completeness property:

$$\mathcal{LP} \vdash F \quad \Leftrightarrow \quad F \textit{ is true under any interpretation .}$$

First order Peano Arithmetic $\mathcal{PA}$ (cf. [8], [9], [30], [43]) is a natural source of such proof systems with Gödel numbers of proofs being a natural instrument of internalizing proofs as terms (natural numbers). In principle any system of proofs with a proof checker operation capable of internalizing its own proofs as terms (cf.[42]) can provide a model for $\mathcal{LP}$. So, the soundness ($\Rightarrow$) does not necessarily refer to the arithmetical models of $\mathcal{LP}$. However, $\mathcal{PA}$ is convenient for establishing the completeness ($\Leftarrow$) of $\mathcal{LP}$. Indeed, given $\mathcal{LP} \nvdash F$ it suffices to deliver one interpretation that makes F false. It turns out that such a *counterinterpretation* can always be chosen from a class of proof systems for $\mathcal{PA}$.

Within Sections 2 and 5 of this paper under Δ_1 and Σ_1 we mean the corresponding classes of arithmetical predicates. We will use $\alpha, \beta, \varphi, \psi, \ldots$ to denote arithmetical formulas unless stated otherwise. We use a new functional symbol $\iota z\varphi(z)$ for each arithmetical formula $\varphi(z)$ and assume that ι-terms could be eliminated by using the small scope convention (cf. [13]). In particular, we assume that $\mathcal{PA}$ contains terms for all primitive recursive functions (cf. [43]), called *primitive recursive terms.* Formulas of the form $f(\vec{x}) = 0$ where $f(\vec{x})$ is a primitive recursive term are *standard primitive recursive formulas. A standard* Σ_1 *formula* is a formula $\exists x\varphi(x, \vec{y})$ where $\varphi(x, \vec{y})$ is a standard primitive recursive formula. An arithmetical formula φ is *provably* Σ_1 if it is provably equivalent in $\mathcal{PA}$ to a standard Σ_1 formula; φ is *provably* Δ_1 iff both φ and $\neg\varphi$ are provably Σ_1. The term $\iota z\varphi(z)$ is called *computable* if $\varphi(z)$ is provably Σ_1. The term $\iota z\varphi(z)$ is *provably total*, if $\mathcal{PA}$ proves that there exists a unique z such that $\varphi(z)$. *A closed computable term* is a computable provably total term $\iota z\varphi(z)$ such that $\varphi(z)$ contains no free variables other than z. Closed computable terms stand for all computable "names" for natural numbers. There is an algorithm which for any closed computable term $\iota z\varphi$ calculates its *value*, i.e. a natural number n such that $\mathcal{PA} \vdash \iota z\varphi = n$. A set of computable terms is closed under substitution. The result of substituting a closed computable term into a Δ_1 formula is again a Δ_1 formula.
A *proof predicate* is a provably Δ_1-formula $Prf(x, y)$ such that for all φ

$$\mathcal{PA} \vdash \varphi \quad \Leftrightarrow \quad \text{for some } n \in \omega \quad Prf(n, \ulcorner\varphi\urcorner) \text{ holds.}$$

A proof predicate *Prf(x,y)* is *normal* if

1) for every proof k the set $T(k) = \{l \mid Prf(k, l)\}$ is finite and the function $\widetilde{T(k)} =$ *the code of* $T(k)$ is provably computable,

2) for every finite set S of theorems of $\mathcal{PA}$, $S \subseteq T(k)$ for some proof k.

For every normal proof predicate Prf there are provably computable terms $m(x, y)$, $a(x, y)$, $c(x)$ such that if s, t are closed computable terms, then $m(s, t)$, $a(s, t)$, $c(\ulcorner t\urcorner)$ are again closed computable terms and for all arithmetical formulas φ, ψ the following formulas are valid:

$Prf(s, \ulcorner\varphi \to \psi\urcorner) \wedge Prf(t, \ulcorner\varphi\urcorner) \to Prf(m(s, t), \ulcorner\psi\urcorner)$

$Prf(s, \ulcorner\varphi\urcorner) \to Prf(a(s, t), \ulcorner\varphi\urcorner), \quad Prf(t, \ulcorner\varphi\urcorner) \to Prf(a(s, t), \ulcorner\varphi\urcorner)$

$Prf(t, \ulcorner\varphi\urcorner) \to Prf(c(\ulcorner t\urcorner), \ulcorner Prf(t, \ulcorner\varphi\urcorner)\urcorner).$

For example, $m(x, y)$ is a natural term for the following computable function:

> *If for some* φ, ψ *both* $Prf(x, \ulcorner\varphi \to \psi\urcorner)$ *and* $Prf(y, \ulcorner\varphi\urcorner)$ *hold, then take the first* z *such that* $Prf(z, \ulcorner\varphi\urcorner)$ *and let* $m(x, y) = z$.

Similarly, $a(x, y)$ is a natural term for the computable function

the least z such that $T(z) \supseteq T(x) \cup T(y)$.

Finally, $c(x)$ is a natural term for the computable function:

> *Given x recover the term f such that $x = \ulcorner f \urcorner$ and calculate $T(f)$. Then for every φ from $T(f)$ compute l_φ such that $Prf(l_\varphi, \ulcorner Prf(f, \ulcorner \varphi \urcorner) \urcorner)$ (such l_φ exists since $Prf(f, \ulcorner \varphi \urcorner)$ is a true Δ_1 formula, therefore provable in $\mathcal{PA}$). Take the first y such that $T(y) \supseteq \{l_\varphi \mid \varphi \in T(f)\}$ and let $c(x) = y$.*

We assume that a normal proof predicate comes with its own fixed choice of such terms $m(x,y)$, $a(x,y)$, $c(x)$. Note, that the natural arithmetical proof predicate *PROOF(x,y)*

> "*x is the code of a derivation* **containing** *a formula with the code y*".

is an example of a normal proof predicate.

2.1 Definition. An arithmetical *interpretation* $*$ of the $\mathcal{LP}$-language has the following parameters:

- a normal proof predicate Prf (with the terms $m(x,y)$, $a(x,y)$, $c(x)$),
- an evaluation of sentence letters by sentences of arithmetic,
- and an evaluation of proof letters by closed recursive terms.

Let $*$ commute with boolean connectives, $(t{\cdot}s)^* \equiv m(t^*, s^*)$, $(t+s)^* \equiv a(t^*, s^*)$, $(!t)^* \equiv c(\ulcorner t^* \urcorner)$, $(t{:}F)^* \equiv Prf(t^*, \ulcorner F^* \urcorner)$. Under any interpretation $*$ a proof polynomial t becomes a closed computable term t^*, an $\mathcal{LP}$-formula F becomes an arithmetical sentence F^*. a formula $(t{:}F)^*$ is always provably Δ_1. Note, that $\mathcal{PA}$ (as well as any theory containing certain finite number of arithmetical axioms, e.g. Robinson's arithmetic) is able to derive any true Δ_1 formula, and thus to derive a negation of any false Δ_1 formula (cf. [30]). For a set X of $\mathcal{LP}$-formulas under X^* we mean the set of all F^*'s such that $F \in X$. Given a constant specification $\mathcal{CS}$, an arithmetical interpretation $*$ is a *$\mathcal{CS}$-interpretation* if all formulas from $\mathcal{CS}^*$ are true (equivalently, are provable in $\mathcal{PA}$). In the functional completeness theorem below we will need one more clause of the definition of $*$: $(\Box F)^* \equiv \exists y Prf(y, \ulcorner F^* \urcorner)$. An $\mathcal{LP}$-formula F is *valid* (with respect to the arithmetical semantics) if the arithmetical formula F^* is true under all interpretations $*$. F is *$\mathcal{CS}$-valid* if F^* is true under all $\mathcal{CS}$-interpretations $*$.

2.2 Theorem. (Arithmetical soundness of $\mathcal{LP}_0$)

1. *If $\mathcal{LP}_0 \vdash F$ then F is valid.*
2. *If $\mathcal{LP}_0 \vdash F$ then $\mathcal{PA} \vdash F^*$ for any interpretation $*$.* (strong soundness)

Proof. The only nontrivial case is establishing the strong soundness of axiom *A1*: $t{:}F \to F$. Let $*$ be an arithmetical interpretation, then $(t{:}F \to F)^*$ is $Prf(t^*, \ulcorner F^* \urcorner) \to F^*$. Consider two possibilities. Either $Prf(t^*, \ulcorner F^* \urcorner)$ is true, in which case the value n of the term t^* is indeed a proof of F^*, thus $\mathcal{PA} \vdash F^*$ and $\mathcal{PA} \vdash (t{:}F \to F)^*$. Otherwise $Prf(t^*, \ulcorner F^* \urcorner)$ is false, in which case being a false Δ_1 formula it is refutable in $\mathcal{PA}$, i.e. $\mathcal{PA} \vdash \neg Prf(t^*, \ulcorner F^* \urcorner)$ and again $\mathcal{PA} \vdash (t{:}F \to F)^*$.
◀

2.3 Corollary. (Arithmetical soundness of $\mathcal{LP}$)
If $\mathcal{LP}(\mathcal{CS}) \vdash F$, then F is $\mathcal{CS}$-valid.

3 Functional completeness of proof polynomials

In this section we show functional completeness for the system of proof polynomials in $\mathcal{LP}$, as another justification for the basic set of operations $\cdot, !, +$. the results in this section are not pertinent to the rest of the paper so the reader may choose to skip this section.

Operations on proofs invariant with respect to the choice of a proof system naturally arise from the notion of admissible rule in arithmetic.

3.1 Definition. A *deterministic admissible rule in $\mathcal{PA}$* is a figure

$$\frac{\vdash C_1, \ldots, \vdash C_n}{\vdash G} \qquad (1)$$

where $C_1, \ldots, C_n, G$ are logical formulas such that for every interpretation $*$, G^* is provable in $\mathcal{PA}$ whenever $C_1{}^*, \ldots, C_n{}^*$ are. A *(non-deterministic) admissible rule in $\mathcal{PA}$ (a.r.)* is a rule of the kind

$$\frac{\mathcal{C}_1 \ \ or \ \ \mathcal{C}_2 \ \ or \ \ \ldots \ \ or \ \ \mathcal{C}_n}{\vdash G}, \qquad (2)$$

where every $\mathcal{C}_i$ is a usual deterministic premise $\vdash C_i^1, \ldots, \vdash C_i^{n_i}$. The meaning of (2) is that for every interpretation $*$ G^* is derivable in $\mathcal{PA}$ whenever for some i, $0 \leq i \leq n$, all $(C_i^1)^*, \ldots, (C_i^{n_i})^*$ are derivable in $\mathcal{PA}$. Every such rule may be regarded as an implicit specification of a proof y of G as a function of proofs x_{ij}'s of C_i^j's. Rule (2) is a *propositional a.r. in a proof format* if all C_i^j's and G are formulas in the language of $\mathcal{LP}$. A propositional a.r. in a proof format (2) is naturally presented by a **valid** formula in the language of $\mathcal{LP}$

with an additional operator $\Box$ representing an implicit existential quantifier over proofs[2]:

$$\bigvee_i \bigwedge_j x_{ij}:C_{ij} \to \Box G \qquad (3)$$

We call such a formula (3) an *invariant operation on proofs.*

3.2 Example. Some examples of *invariant operations on proofs*:

i) $x:(F \to G) \wedge y:F \to \Box G$,
ii) $x:F \to \Box x:F$,
iii) $x:F \wedge y:G \to \Box(F \wedge G)$,
iv) $x:F \vee y:G \to \Box(F \vee G)$.

For each of these examples there is a proof polynomial p realizing the operator $\Box$ in such a way that instantiating p inside $\Box$ gives a formula derivable in $\mathcal{LP}$:

i) $\mathcal{LP} \vdash x:(F \to G) \wedge y:F \to (x \cdot y):G$,
ii) $\mathcal{LP} \vdash x:F \to !x:x:F$,
iii) $\mathcal{LP} \vdash x:F \wedge y:G \to t(x,y):(F \wedge G)$, (from Example 1.3)
iv) $\mathcal{LP} \vdash x:F \vee y:G \to (ax+by):(F \vee G)$. (from Example 1.4)

The following theorem demonstrates that proof existence in any absolute operation on proofs can be instantiated with a specific proof polynomial.

3.3 Theorem. *For any invariant operation on proofs*

$$\bigvee_i \bigwedge_j x_{ij}:C_{ij} \to \Box G$$

one can construct a proof polynomial $p(\vec{x})$ *such that*

$$\mathcal{LP} \vdash \bigvee_i \bigwedge_j x_{ij}:C_{ij} \to p(\vec{x}):G.$$

Proof. Let (2) be an invariant operation on proofs, and let us denote

$$\bigvee_i \bigwedge_j x_{ij}:C_{ij}$$

[2] $(\Box F)^* \equiv \exists y Prf(y, \ulcorner F^* \urcorner)$.

by C. Since $\Box G \to G$ is valid (under all arithmetical interpretations), the formula $C \to G$ is also valid. By the completeness theorem for $\mathcal{LP}$ (see Section 5), $\mathcal{LP} \vdash C \to G$. By Lifting 1.2 and Deduction, for a fresh variable y there is a proof polynomial $t(y)$ such that

$$\mathcal{LP} \vdash y{:}C \to t(y){:}G. \qquad (4)$$

3.4 Lemma. *For any formulas A, B one can construct a proof polynomial $u(x,y)$ such that*

$$\mathcal{LP} \vdash x{:}A \wedge y{:}B \to u(x,y){:}(x{:}A \wedge y{:}B).$$

Indeed, $\mathcal{LP} \vdash x{:}A \to !x{:}x{:}A$, $\mathcal{LP} \vdash y{:}B \to !y{:}y{:}B$. By 1.3,

$$\mathcal{LP} \vdash !x{:}x{:}A \wedge !y{:}y{:}B \to t(!x,!y){:}(x{:}A \wedge y{:}B)$$

for an appropriate t, thus

$$\mathcal{LP} \vdash x{:}A \wedge y{:}B \to t(!x,!y){:}(x{:}A \wedge y{:}B).$$

3.5 Lemma. *For all formulas A, B there exists a proof polynomial $v(x,y)$ such that*

$$\mathcal{LP} \vdash x{:}A \vee y{:}B \to v(x,y){:}(x{:}A \vee y{:}B).$$

Similar to the proof of 3.4, using 1.4.

3.6 Lemma. *One can construct a proof polynomial $s(\vec{x})$ such that*

$$\mathcal{LP} \vdash C \to s(\vec{x}){:}C.$$

Proof. This lemma is proved by a straightforward induction on the formula

$$C = \bigvee_i \bigwedge_j x_{ij}{:}C_{ij}$$

with the use of 3.4 and 3.5.
◀

From (4), we have $\mathcal{LP} \vdash s : C \to t(s) : G$, and thus from 3.6 we get the desired result $\mathcal{LP} \vdash C \to t(s){:}G$. This concludes the proof of 3.3.
◀

3.7 Comment. Whereas (2) requires all three proof connectives, the realization of deterministic operations of the form (1) does not require "+".

3.8 Comment. We demonstrated that every invariant propositional operation on proofs can be represented by a corresponding proof polynomial. Thus the basic operations $\cdot, !, +$ to some extent play for the proofs a role similar to that played by the boolean connectives in classical logic where every truth function (total function from $\{0,1\}^n$ to $\{0,1\}$) is represented by a boolean polynomial.

4 A sequent formulation of Logic of Proofs

Throughout Sections 4 - 7 by *sequent* we mean a pair $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of $\mathcal{LP}$-formulas. Axioms of $\mathcal{LPG}_0$ are sequents of the form $F \Rightarrow F$ and $\bot \Rightarrow$. Along with the usual Gentzen sequent rules of classical propositional logic, including the cut rule (cf. the system **G1c** from [45]), the system $\mathcal{LPG}_0$ contains the rules

$$\frac{A, \Gamma \Rightarrow \Delta}{t{:}A, \Gamma \Rightarrow \Delta}\,(: \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, t{:}A}{\Gamma \Rightarrow \Delta, !t{:}t{:}A}\,(\Rightarrow !)$$

$$\frac{\Gamma \Rightarrow \Delta, t{:}A}{\Gamma \Rightarrow \Delta, (t+s){:}A}\,(\Rightarrow +) \qquad \frac{\Gamma \Rightarrow \Delta, t{:}A}{\Gamma \Rightarrow \Delta, (s+t){:}A}\,(\Rightarrow +)$$

$$\frac{\Gamma \Rightarrow \Delta, s{:}(A \to B) \qquad \Gamma \Rightarrow \Delta, t{:}A}{\Gamma \Rightarrow \Delta, (s \cdot t){:}B}\,(\Rightarrow \cdot)$$

As will follow from the proof of 5.1 the rule $(\Rightarrow \cdot)$ for $\mathcal{LPG}_0$ (but not for $\mathcal{LPG}$) can in fact be limited by the condition that $A \to B$ must occur in Γ, Δ, without losing any provable sequents

The system $\mathcal{LPG}$ is $\mathcal{LPG}_0$ plus the rule

$$\frac{}{\Rightarrow c{:}\mathbf{A}}\,(\Rightarrow c),$$

where **A** is an axiom $A0$ - $A4$ from Section 1, and c is a proof constant.

Under $\mathcal{LPG}^-$ and $\mathcal{LPG}_0^-$ we mean the corresponding systems without the rule Cut.

4.1 Theorem. $\mathcal{LPG}_0 \vdash \Gamma \Rightarrow \Delta$ *iff* $\mathcal{LP}_0 \vdash \bigwedge\Gamma \rightarrow \bigvee\Delta$, $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$ *iff* $\mathcal{LP} \vdash \bigwedge\Gamma \rightarrow \bigvee\Delta$.

The proof proceeds by a straightforward induction both ways.

4.2 Corollary. $\mathcal{LP}(\mathcal{CS}) \vdash F$ *iff* $\mathcal{LPG}_0 \vdash \mathcal{CS} \Rightarrow F$

4.3 Definition. The sequent $\Gamma \Rightarrow \Delta$ is *saturated* if

1. $A \rightarrow B \in \Gamma$ implies $B \in \Gamma$ or $A \in \Delta$,
2. $A \rightarrow B \in \Delta$ implies $A \in \Gamma$ and $B \in \Delta$[3],
3. $t{:}A \in \Gamma$ implies $A \in \Gamma$,
4. $!t{:}t{:}A \in \Delta$ implies $t{:}A \in \Delta$,
5. $(s+t){:}A \in \Delta$ implies $s{:}A \in \Delta$ and $t{:}A \in \Delta$
6. $(s \cdot t) : B \in \Delta$ implies *for each* $X \rightarrow B$ *occurring as a subformula in* Γ, Δ *either* $s{:}(X \rightarrow B) \in \Delta$ *or* $t{:}X \in \Delta$.

4.4 Lemma. (Saturation lemma) *Suppose* $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. *Then there exists a saturated sequent* $\Gamma' \Rightarrow \Delta'$ *such that*

1. $\Gamma \subseteq \Gamma'$, $\Delta \subseteq \Delta'$,
2. $\Gamma' \Rightarrow \Delta'$ *is not derivable in* $\mathcal{LPG}_0^-$.

Proof. A saturated sequent is obtained by the following *Saturation Algorithm* $\mathcal{SAT}$. Given $\Gamma \Rightarrow \Delta$, for each formula S from $\Gamma \cup \Delta$ nondeterministically try to perform one of the following steps. If none of the clauses 1 - 6 is applicable terminate with success.

1. if $S = (A \rightarrow B) \in \Gamma$, then put A into Δ or B into Γ,
2. if $S = (A \rightarrow B) \in \Delta$, then put A into Γ and B into Δ,
3. if $S = t{:}A \in \Gamma$, then put A into Γ,
4. if $S = !t{:}t{:}A \in \Delta$, then put $t{:}A$ into Δ,
5. if $S = (s+t){:}A \in \Delta$, then put both $s{:}A$ and $t{:}A$ into Δ,
6. if $S = (s \cdots t){:}B \in \Delta$, then for each $X_1, \ldots X_n$ such that $X_i \rightarrow B$ is a subformula in Γ, Δ put either $s{:}(X_i \rightarrow B)$ or $t{:}X_i$ into Δ,
7. if $\Gamma \cap \Delta \neq \emptyset$ or $\bot \in \Gamma$, then backtrack. If backtracked to the root node terminate with failure.

The Saturation Algorithm $\mathcal{SAT}$ terminates. Indeed, $\mathcal{SAT}$ is finitely branching and each non-backtracking step breaks either a subformula of $\Gamma \Rightarrow \Delta$ or a formula of the type $t{:}F$, where both t and F occur in $\Gamma \Rightarrow \Delta$. There are only finitely many of those formulas, which guarantees

[3]The clauses concerning other boolean connectives are optional.

termination. Moreover, $\mathcal{SAT}$ terminates with success. Indeed, otherwise $\mathcal{SAT}$ terminates at the root node $\Gamma \Rightarrow \Delta$ of the computation tree with all the possibilities exhausted and no way to backtrack. Then the computation tree $\mathcal{T}$ of $\mathcal{SAT}$ contains the sequent $\Gamma \Rightarrow \Delta$ at the root, and $\mathcal{LPG}_0$ axioms at the leaf nodes. By a standard induction on the depth of a node in $\mathcal{T}$ one can prove, that every sequent in $\mathcal{T}$ is derivable in $\mathcal{LPG}_0^-$, which contradicts the assumption that $\mathcal{LPG}_0^- \not\vdash \Gamma \Rightarrow \Delta$. The nodes corresponding to the steps 1 – 4 and 6 are trivial. Let us consider a node, which corresponds to 5. Such a node is labelled by a sequent $\Pi \Rightarrow \Theta, st{:}B$, and its children are 2^n sequents of the form $\Pi \Rightarrow \Theta, st{:}B, Y_1^\sigma, \ldots, Y_n^\sigma$, where $\sigma = (\sigma_1 \ldots, \sigma_n)$ is an n-tuple of 0's and 1's, and

$$Y_i^\sigma = \begin{cases} s{:}(X_i \to B), & \text{if } \sigma_i = 0 \\ t{:}X_i, & \text{if } \sigma_i = 1. \end{cases}$$

Here $X_1, \ldots, X_n$ is the list of all formulas such that $X_i \to B$ is a subformula of $\Gamma \Rightarrow \Delta$. By the induction hypothesis all the child sequents are derivable in $\mathcal{LPG}_0^-$. In particular, among them there are 2^{n-1} pairs of sequents of the form $\Pi \Rightarrow \Theta', s{:}(X_1 \to B)$ and $\Pi \Rightarrow \Theta', t{:}X_1$. To every such pair apply the rule $(\Rightarrow \cdot)$ to obtain $\Pi \Rightarrow \Theta'$ (we assume, that $st : B \in \Theta'$. The resulting 2^{n-1} sequents are of the form $\Pi \Rightarrow \Theta, st{:}B, Y_2^\sigma, \ldots, Y_n^\sigma$. After we repeat this procedure $n-1$ more times we end up with the sequent $\Pi \Rightarrow \Theta, st{:}B$, which is thus derivable in $\mathcal{LPG}_0^-$.

◀

Note, that in a saturated sequent $\Gamma \Rightarrow \Delta$ the set Γ is closed under the rules $t : X/X$ and $X \to Y, X/Y$.

4.5 Lemma. *For each saturated sequent $\Gamma \Rightarrow \Delta$ not derivable in $\mathcal{LPG}_0^-$ there is a set of $\mathcal{LP}$-formulas $\widetilde{\Gamma}$ (a completion of $\Gamma \Rightarrow \Delta$) such that*

1. $\widetilde{\Gamma}$ is a provably decidable set, for each term t the set $I(t) = \{X \mid t{:}X \in \widetilde{\Gamma}\}$ is finite and a function from a code[4] of t to a code[5] of $I(t)$ is provably computable,

2. $\Gamma \subseteq \widetilde{\Gamma}$, $\Delta \cap \widetilde{\Gamma} = \emptyset$,

3. if $t{:}X \in \widetilde{\Gamma}$, then $X \in \widetilde{\Gamma}$,

4. if $s{:}(X \to Y) \in \widetilde{\Gamma}$ and $t{:}X \in \widetilde{\Gamma}$, then $(s \cdot t){:}Y \in \widetilde{\Gamma}$,

5. if $t{:}X \in \widetilde{\Gamma}$, then $!t{:}t{:}X \in \widetilde{\Gamma}$,

6. if $t{:}X \in \widetilde{\Gamma}$ and s is a proof polynomial, then $(t+s){:}X \in \widetilde{\Gamma}$ and $(s+t){:}X \in \widetilde{\Gamma}$.

Proof. We describe a *completion algorithm* $\mathcal{COM}$ that produces a series of finite sets of $\mathcal{LP}$-formulas $\Gamma_0, \Gamma_1, \Gamma_2 \ldots$. Let $\Gamma_0 = \Gamma$.

For each natural number $i > 1$ let $\mathcal{COM}$ do the following:

[4]For example, a Gödel number of t.

[5]For example, a canonical number of a finite set of Gödel numbers of the formulas from $I(t)$.

if $i = 3k$, then $\mathcal{COM}$ puts

$$\Gamma_{i+1} = \Gamma_i \bigcup_{s,t} \{(s \cdot t) : Y \mid s : (X \to Y), t : X \in \Gamma_i\},$$

if $i = 3k + 1$, then $\mathcal{COM}$ puts

$$\Gamma_{i+1} = \Gamma_i \bigcup_{t} \{!t : t : X \mid t : X \in \Gamma_i\},$$

if $i = 3k + 2$, then $\mathcal{COM}$ puts

$$\Gamma_{i+1} = \Gamma_i \bigcup_{s,t} \{(s + t) : X, (t + s) : X \mid t : X \in \Gamma_i, |s| < i.\}^6$$

Put

$$\widetilde{\Gamma} = \bigcup_{i} \Gamma_i.$$

1. It is easy to see that at step $i > 0$ $\mathcal{COM}$ produces formulas of the form $t : X$ with the length of t greater than i. This observation secures the decidability of $\widetilde{\Gamma}$. Indeed, given a formula F of length n wait until step $i = n$ of $\mathcal{COM}$; $F \in \Gamma_n$ iff $F \in \widetilde{\Gamma}$. Similar argument establishes the decidability of $I(t)$.

Note that all Γ_i's are closed under the rules $t : X/X$ and $X \to Y, X/Y$. The closure under $X \to Y, X/Y$ is obvious, since it holds for Γ_0 and $\mathcal{COM}$ does not add any new formulas of the form $X \to Y$. To establish the case of $t : X/X$ we use an induction on i. Suppose Γ_i is closed under $t : X/X$. If $i = 3k$, then all new formulas at this step are of the form $(s \cdot t) : Y$ with $s : (X \to Y), t : X \in \Gamma_i$. By the IH, $X \to Y, X \in \Gamma_i$ and $Y \in \Gamma_i$, thus $Y \in \Gamma_{i+1}$. In particular, it secures property 3.

2. The inclusion $\Gamma \subseteq \widetilde{\Gamma}$ is clear. Suppose i is the least number such that there exists a formula $F \in \Gamma_i \cap \Delta$. F cannot be from Γ_0, since $\Gamma_0 = \Gamma$ and Δ are disjoint. Thus F has been introduced by $\mathcal{COM}$ at step $i > 0$. Suppose $i - 1 = 3k$. Then $F = (s \cdot t) : Y$ and for some formula X we have $s : (X \to Y), t : X \in \Gamma_{i-1}$. In such a case $X \to Y \in \Gamma_0 = \Gamma$, since a formula with the implication as the principal connective cannot first appear in Γ_j with $j > 0$. Suppose $(s \cdot t) : Y \in \Delta$. Then by the saturation properties of the sequent $\Gamma \Rightarrow \Delta$, either $s : (X \to Y) \in \Delta$ or $t : X \in \Delta$. In either case $\Gamma_{i-1} \cup \Delta \neq \emptyset$. The remaining clauses follow easily from the definitions and saturation properties of the sequent $\Gamma \Rightarrow \Delta$.

4., 5., 6. are guaranteed by the definition of $\mathcal{COM}$. Indeed, if some *if* condition is fulfilled, then it occurs at step i and $\mathcal{COM}$ necessarily puts the *then* formula into Γ_{i+3} at the latest.
◀

[6] $|s|$ is the length of s, i.e. a total number of variables, constants, and functional symbols in s.

5 Consolidated completeness theorem

In this section we establish completeness and cut elimination theorems for the Logic of Proofs.

5.1 Theorem. *The following are equivalent*

1. $\mathcal{LPG}_0^- \vdash \Gamma \Rightarrow \Delta$,
2. $\mathcal{LPG}_0 \vdash \Gamma \Rightarrow \Delta$,
3. $\mathcal{LP}_0 \vdash \bigwedge \Gamma \to \bigvee \Delta$,
4. *for every interpretation* $*$ $\mathcal{PA} \vdash (\bigwedge \Gamma \to \bigvee \Delta)^*$,
5. *for every interpretation* $*$ $(\bigwedge \Gamma \to \bigvee \Delta)^*$ *is true.*

Proof. The steps from 1 to 2 and from 4 to 5 are trivial. The step from 2 to 3 follows from 4.1, from 3 to 4 follows from 2.2. The only remaining step is thus from 5 to 1. We assume "not 1" and establish "not 5". Suppose $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. Our aim now will be to construct an interpretation $*$ such that $(\bigwedge \Gamma \to \bigvee \Delta)^*$ is false (in the standard model of arithmetic).

Perform a saturation procedure to get a saturated sequent $\Gamma' \Rightarrow \Delta'$ (4.4), and then make a completion to get a set of formulas $\widetilde{\Gamma}'$ (4.5).

We define the desired interpretation $*$ on sentence letters S_i, proof variables x_j and proof constants a_j first. As usual, $\ulcorner t \urcorner$ denote the Gödel number of t. Put

$$S_i^* = \begin{cases} i+1 = i+1, & \text{if } S_i \in \widetilde{\Gamma}' \\ i+1 = 0, & \text{if } S_i \notin \widetilde{\Gamma}', \end{cases} \qquad x_j{}^* = \ulcorner x_j \urcorner, \qquad a_j{}^* = \ulcorner a_j \urcorner.$$

The remaining parts of $*$ are constructed by a multiple arithmetical fixed point equation. Let $(PROOF, \otimes, \oplus, \Uparrow)$ be a standard nondeterministic proof predicate from Section 2, with $\otimes$ standing for application, $\oplus$ for choice and $\Uparrow$ for proof checker operations on proofs associated with $PROOF$. For technical convenience and without loss of generality we assume that $PROOF(\ulcorner t \urcorner, k)$ is false for any $\mathcal{LP}$-term t and any $k \in \omega$.

Let $\varphi(x, \vec{y})$ be a provably Σ_1 formula, i.e. $\varphi(x, \vec{y})$ is provably equivalent to $\exists z \psi(x, \vec{y}, z)$ for some standard primitive recursive formula $\psi(x, \vec{y}, z)$. Under $\mu x \varphi(x, \vec{y})$ we mean a natural term for a computable function $f(\vec{y})$:

Given $\vec{y}$ find a tuple $(u, \vec{y}, z)$ with the least number such that $\psi(u, \vec{y}, z)$. Let $f(\vec{y}) = u$.

In what follows $*$ is based on a normal proof predicate Prf,

$$m(x, y) = \mu z M(x, y, z), \quad a(x, y) = \mu z A(x, y, z), \quad c(x) = \mu z C(x, z),$$

where Prf , $M(x, y, z)$, $A(x, y, z)$, $C(x, z)$ are determined by a fixed point equation FPE below. Note, that $\ulcorner B^* \urcorner$ can be calculated in a primitive recursive way from

$$\ulcorner Prf(x, y) \urcorner, \ \ulcorner M(x, y, z) \urcorner, \ \ulcorner C(x, z) \urcorner, \ulcorner A(x, y, z) \urcorner$$

for any subformula B from $\widetilde{\Gamma'} \cup \Delta'$.

By the arithmetical fixed point argument there exist arithmetical formulas $Prf(x,y)$, $M(x,y,z)$, $A(x,y,z)$, $C(x,z)$ such that $\mathcal{PA}$ proves the following *fixed point equation (FPE)*:

$$
\begin{array}{lcl}
Prf(x,y) & \leftrightarrow & PROOF(x,y) \ \vee \\
 & & \text{“}x = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-term } t\text{”} \wedge y = \ulcorner B^* \urcorner \wedge \text{“}B \in I(t)\text{”}.
\end{array}
$$

$$
\begin{array}{lcl}
M(x,y,z) & \leftrightarrow & \textit{if } x = \ulcorner s \urcorner \textit{ and } y = \ulcorner t \urcorner \textit{ for some terms } s,t, \textit{ then } z = \ulcorner st \urcorner , \\
 & & \\
 & & \textit{if } x = \ulcorner s \urcorner \textit{ and } y \neq \ulcorner t \urcorner \textit{ for any term } t, \textit{ then recover } I(s), \\
 & & \textit{put } z = \mu w(\bigwedge\{PROOF(w, \ulcorner B^* \urcorner) \mid B \in I(s)\}) \otimes y, \\
 & & \\
 & & \textit{if } y = \ulcorner t \urcorner \textit{ and } x \neq \ulcorner s \urcorner \textit{ for any term } s, \textit{ then recover } I(t), \\
 & & \textit{put } z = x \otimes \mu w(\bigwedge\{PROOF(w, \ulcorner B^* \urcorner) \mid B \in I(t)\}), \\
 & & \\
 & & z = x \otimes y, \textit{ else.}
\end{array}
$$

$$
\begin{array}{lcl}
C(x,z) & \leftrightarrow & \textit{if } x = \ulcorner t^* \urcorner, \textit{ then } z = \ulcorner !t \urcorner , \\
 & & z = \Uparrow x, \textit{ else.}
\end{array}
$$

$$
\begin{array}{lcl}
A(x,y,z) & \leftrightarrow & \textit{if } x = \ulcorner s \urcorner, \ y = \ulcorner t \urcorner \textit{ for some terms } s,t, \textit{ then } z = \ulcorner s + t \urcorner , \\
 & & \\
 & & \textit{if } x = \ulcorner s \urcorner \textit{ and } y \neq \ulcorner t \urcorner \textit{ for any term } t, \textit{ then recover } I(s), \\
 & & \textit{put } z = \mu w(\bigwedge\{PROOF(w, \ulcorner B^* \urcorner) \mid B \in I(s)\}) \oplus y, \\
 & & \\
 & & \textit{if } y = \ulcorner t \urcorner \textit{ and } x \neq \ulcorner s \urcorner \textit{ for any term } s, \textit{ then recover } I(t), \\
 & & \textit{put } z = x \oplus \mu w(\bigwedge\{PROOF(w, \ulcorner B^* \urcorner) \mid B \in I(t)\}), \\
 & & \\
 & & z = x \oplus y, \textit{ else.}
\end{array}
$$

Here $\mu w(\bigwedge\{PROOF(w, \ulcorner B^* \urcorner) \mid B \in I(t)\})$ is a natural computable ι-term corresponding to the following computable procedure:

> *“Given $\ulcorner t \urcorner$ take a code of $I(t)$ and calculate a code of $I^*(t)$, which is a finite set of arithmetical translations of $\mathcal{LP}$ formulas from $I(t)$. Set w equal to the least number which is a proof of all formulas from $I^*(t)$ in the sense of PROOF. Put w=0, if $I(t)$ is empty. ”*

The convergence of $\mu w(\bigwedge\{PROOF(w, \ulcorner B^* \urcorner) \mid B \in I(t)\})$ will be established in 5.7. By *FPE* it is immediate that *Prf* is a provably Δ_1-formula and if $\mathcal{PA} \vdash \psi$, then $Prf(k, \ulcorner \psi \urcorner)$ for some

$k \in \omega$. Also by *FPE*, each of the formulas M, A, and C is provably Σ_1, and thus the terms $m(x,y)$, $a(x,y)$ and $c(x)$ are computable.

5.2 Lemma. *$\mathcal{PA} \vdash t^* = \ulcorner t \urcorner$ for any term t.*

Indeed, according to *FPE*

$$(st)^* = m(s^*, t^*) = m(\ulcorner s \urcorner, \ulcorner t \urcorner) = \ulcorner st \urcorner.$$

The same holds for $(s+t)^*$ and $!t^*$.

5.3 Corollary. *$*$ is injective on formulas and terms from $\widetilde{\Gamma}' \cup \Delta'$.*

5.4 Corollary. *X^* is provably Δ_1 for any X occurring as a subformula in $\widetilde{\Gamma}' \cup \Delta'$.*

Indeed, if X is atomic, then X is Δ_1 by the definition of $*$. If X is $t{:}Y$, then

$$(t{:}Y)^* \quad = \quad Prf(t^*, \ulcorner Y^* \urcorner),$$

and since

$$\mathcal{PA} \vdash Prf(t^*, \ulcorner Y^* \urcorner) \leftrightarrow Prf(\ulcorner t \urcorner, \ulcorner Y^* \urcorner)$$

and $Prf(\ulcorner t \urcorner, \ulcorner Y^* \urcorner)$ is provably Δ_1, then $(t{:}Y)^*$ surely is Δ_1. Since Δ_1 is closed under boolean connectives for each X occurring as a subformula in $\widetilde{\Gamma}' \cup \Delta'$, X^* is provably Δ_1.

5.5 Lemma. *If $X \in \widetilde{\Gamma}'$, then $\mathcal{PA} \vdash X^*$, if $X \in \Delta'$, then $\mathcal{PA} \vdash \neg X^*$.*

Proof. By induction on the length of X. Base case, i.e. X is atomic or $X = t{:}Y$. If X is atomic, then X^* is true iff $X \in \widetilde{\Gamma}'$ by definition and since $\widetilde{\Gamma}'$ and Δ' are disjoint. If $X = t{:}Y$ and $t{:}Y \in \widetilde{\Gamma}'$, then $\mathcal{PA} \vdash$ "$Y \in I(t)$" and $\mathcal{PA} \vdash (t{:}Y)^*$ by *FPE*. If $t{:}Y \in \Delta'$, then "$Y \in I(t)$" is false, since $\widetilde{\Gamma}'$ and Δ' are disjoint. The formula $PROOF(t^*, \ulcorner Y^* \urcorner)$ is also false since $t^* = \ulcorner t \urcorner$ and $PROOF(\ulcorner t \urcorner, k)$ is false by assumption. Thus $(t{:}Y)^*$ is false by *FPE*. The induction steps corresponding to boolean connectives are standard and based on the saturation properties of $\Gamma' \Rightarrow \Delta'$. For example, let $X = Y \to Z$ and $Y \to Z \in \widetilde{\Gamma}'$. Then $Y \to Z \in \Gamma'$, and by definition 4.3 $Y \in \Gamma'$ or $Z \in \Delta'$. By the induction hypothesis, Y^* is true or Z^* is false, and thus $(Y \to Z)^*$ is true, etc.
◀

5.6 Lemma. *$\mathcal{PA} \vdash \varphi \quad \Leftrightarrow \quad Prf(n, \ulcorner \varphi \urcorner)$ for some $n \in \omega$.*

Proof. It remains to establish ($\Leftarrow$). From *FPE* it is clear that

$$Prf(n,\ulcorner\psi\urcorner) \;\Rightarrow\; \text{“}PROOF(n,\ulcorner\psi\urcorner) \text{ or } \psi = B^* \text{ for some } B \text{ such that } t{:}B \in \widetilde{\Gamma'}\text{”}.$$

In the latter case $B \in \widetilde{\Gamma'}$ by the saturation property of $\widetilde{\Gamma'}$, and $\mathcal{PA} \vdash B^*$ by 5.5.
◀

5.7 Lemma. *For any closed computable terms s,t the terms $m(s,t)$, $a(s,t)$, $c(t)$ are closed and computable.*

Proof. It suffices to establish, that for any proof polynomial t the arithmetical term $\mu w(\bigwedge\{PROOF(w,\ulcorner B^*\urcorner) \mid B \in I(t)\})$ is closed and computable. For that it suffices to check that $\mathcal{PA} \vdash \exists w(\bigwedge\{PROOF(w,\ulcorner B^*\urcorner) \mid B \in I(t)\})$. Argue in $\mathcal{PA}$. Given $\ulcorner t\urcorner$ take a code of $I(t)$ and a code of $I^*(t)$, which is a finite set of arithmetical translations of $\mathcal{LP}$ formulas from $I(t)$. By 5.5 $\mathcal{PA}$ proves all arithmetical formulas from $I^*(t)$, thus $\mathcal{PA} \vdash \bigwedge\{PROOF(k,\ulcorner B^*\urcorner) \mid B \in I(t)\}$ for some natural number k.
◀

5.8 Lemma. *The normality conditions for Prf are fulfilled.*

Proof. Normality checklist:

1. *Prf* is provably Δ_1 (follows from *FPE*). Conditions on the arithmetical terms $m(x,y)$, $a(x,y)$ and $c(x)$ are fulfilled by *FPE* and 5.7.

2. for any finite set S of arithmetical formulas $\mathcal{PA} \vdash S$ if and only if $Prf(n,S)$ holds for some natural n (follows from *FPE* and 5.6).

3. for each n the set of formulas $\{\varphi \mid Prf(n,\ulcorner\varphi\urcorner)\}$ is finite. Indeed, either n is not a number of an $\mathcal{LP}$-term, then use the normality of *PROOF*. Otherwise use the finiteness of $I(t)$ and the injectivity of $*$.

◀

Let us finish the proof of the concluding “not 1 implies not 5” part of 5.1. Given a sequent $\Gamma \Rightarrow \Delta$ not provable in $\mathcal{LPG}_0^-$ we have constructed an interpretation $*$ such that Γ^* are all true, and Δ^* are all false in the standard model of arithmetic (5.5). Therefore, $(\bigwedge\Gamma \to \bigvee\Delta)^*$ is false.
◀

5.9 Corollary. *$\mathcal{LP}_0$ is decidable.*

Indeed, given an $\mathcal{LP}$-formula F run the saturation algorithm $\mathcal{SAT}$ on a sequent $\Rightarrow F$. If $\mathcal{SAT}$ fails, then $\mathcal{LP}_0 \vdash F$. Otherwise, $\mathcal{LP}_0 \nvdash F$.

5.10 Corollary. (Completeness of $\mathcal{LP}$ with respect to the provability semantics)

$$\begin{array}{lll} \mathcal{LP}(\mathcal{CS}) \vdash F & \Leftrightarrow & \mathcal{PA} \vdash F^* \textit{ for any } \mathcal{CS}\textit{-interpretation } *. \\ & \Leftrightarrow & F^* \textit{ is true for any } \mathcal{CS}\textit{-interpretation } *. \end{array}$$

5.11 Corollary. (Cut elimination in $\mathcal{LP}$.) *Every sequent derivable in $\mathcal{LPG}$ can be derived without the cut rule.*

Proof. Let $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$. For each use of the rule $\dfrac{}{\Rightarrow c{:}\mathbf{A}}$ replace it by the axiom sequent $c{:}\mathbf{A} \Rightarrow c{:}\mathbf{A}$, and add $c{:}\mathbf{A}$ to the antecedents of all the sequents below the given node. The result will be a legitimate derivation in $\mathcal{LPG}_0$ of the sequent $\mathcal{CS}, \Gamma \Rightarrow \Delta$ for a constants specification $\mathcal{CS}$ consisting of all those $c:\mathbf{A}$'s. By 5.1 there exists a cut-free derivation $\mathcal{D}$ of $\mathcal{CS}, \Gamma \Rightarrow \Delta$ in $\mathcal{LPG}_0$. Without loss of generality we may assume that every formula $c:\mathbf{A}$ from $\mathcal{CS}$ there has been introduced in $\mathcal{D}$ by an axiom $c:\mathbf{A} \Rightarrow c:\mathbf{A}$. Two other options: the weakening rule, or the rule $(: \Rightarrow)$ can be eliminated. The weakenings can be easily removed without any harm to a derivation. By induction on the cardinality of $\mathcal{CS}$ we show that there exists a cut-free derivation of $\mathcal{CS}', \Gamma \Rightarrow \Delta$ in $\mathcal{LPG}_0$ such that $\mathcal{CS}' \subseteq \mathcal{CS}$ and no occurrence of a formula $c:\mathbf{A}$ from $\mathcal{CS}'$ has been introduced in this derivation by the rule $(: \Rightarrow)$. The base case corresponds to an axiom and is trivial. Every use of the rule $(: \Rightarrow)$

$$\dfrac{\begin{array}{c}\mathcal{D}' \\ \mathbf{A}, \Gamma' \Rightarrow \Delta'\end{array}}{c{:}\mathbf{A}, \Gamma' \Rightarrow \Delta'} \quad \text{transforms into} \quad \dfrac{\begin{array}{cc}\mathcal{D}'_0 & \mathcal{D}' \\ \Rightarrow \mathbf{A} & \mathbf{A}, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma' \Rightarrow \Delta'}$$

The end sequent $\Gamma' \Rightarrow \Delta'$ of the latter derivation is one formula from $\mathcal{CS}$ less and, by the IH admits a cut-free derivation without an introduction of formulas from $\mathcal{CS}$ by the rule $(: \Rightarrow)$. Finally, in a cut-free derivation of $\mathcal{CS}, \Gamma \Rightarrow \Delta$ in $\mathcal{LPG}_0$ without weakenings and $(: \Rightarrow)$ for formulas from $\mathcal{CS}$ replace each axiom $c:\mathbf{A} \Rightarrow c:\mathbf{A}$ for $c:\mathbf{A} \in \mathcal{CS}$ by the rule $(\Rightarrow c)$ introducing $\Rightarrow c:\mathbf{A}$. The result will be a derivation in $\mathcal{LP}$ of the sequent $\Gamma \Rightarrow \Delta$.

Of course, cut elimination for $\mathcal{LP}$ can be also established by a direct system of reductions (cf. Section 8 of this paper). However, here we have got the cut elimination theorem almost for free, as a side product of the arithmetical completeness theorem.

◀

5.12 Comment. Decidability of $\mathcal{LP}$ follows from the results of [33]. This fact can also be obtained from the cut elimination property of $\mathcal{LP}$ (Corollary 5.11) as as exercise.

6 Realization of modal and intuitionistic logics

It is easy to see that a forgetful projection of $\mathcal{LP}$ is correct w.r.t. $\mathcal{S}4$. Let F^o be the result of substituting $\Box X$ for all occurrences of $t:X$ in F, and $\Gamma^o = \{F^o \mid F \in \Gamma\}$ for any set Γ of $\mathcal{LP}$-formulas.

6.1 Lemma. *If $\mathcal{LP} \vdash F$, then $\mathcal{S}4 \vdash F^o$.*

Proof. This is a straightforward induction on a derivation in $\mathcal{LP}$.
◀

The goal of the current section is to establish the converse, namely that $\mathcal{LP}$ suffices to realizate any $\mathcal{S}4$ theorem. By an *$\mathcal{LP}$-realization* of a modal formula F we mean an assignment of proof polynomials to all occurrences of the modality in F, F^r is the image of F under a realization r. Positive and negative occurrences of modality in a formula and a sequent are defined in the usual way.

1. An indicated occurrence of $\Box$ in $\Box F$ is positive.

2. Any occurrence of $\Box$ in $G \to F$, $G \wedge F$, $F \wedge G$, $G \vee F$, $F \vee G$, $\Box F$ and $\Gamma \Rightarrow \Delta, F$ has the same polarity as the corresponding occurrence of $\Box$ in F.

3. Any occurrence of $\Box$ in $\neg F$, $F \to G$ and $F, \Gamma \Rightarrow \Delta$ has a polarity opposite to that of the corresponding occurrence of $\Box$ in F.

In a provability context $\Box F$ is intuitively understood as "*there exists a proof x of F*". After a skolemization, all negative occurrences of $\Box$ produce arguments of Skolem functions, while positive ones give functions of those arguments. For example, $\Box A \to \Box B$ should be read informally as

$$\exists x \text{ “ } x \text{ is a proof of } A\text{” } \to \exists y \text{ “ } y \text{ is a proof of } B\text{”},$$

with the Skolem form

$$\text{“ } x \text{ is a proof of } A\text{” } \to \text{ “ } f(x) \text{ is a proof of } B\text{”}.$$

The following definition captures this feature. A realization r is *normal* if all negative occurrences of $\Box$ are realized by proof letters.

6.2 Theorem. *If $\mathcal{S}4 \vdash F$, then $\mathcal{LP} \vdash F^r$ for some normal realization r.*

Proof. Consider a cut-free sequent formulation of $\mathcal{S}4$, with sequents $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of modal formulas. Axioms are sequents of the form $S \Rightarrow S$, where S is a sentence letter and $\bot \Rightarrow$. Along with the usual structural rules and rules introducing boolean connectives there are two proper modal rules (cf.[45]):

$$\frac{A, \Gamma \Rightarrow \Delta}{\Box A, \Gamma \Rightarrow \Delta}\,(\Box \Rightarrow) \qquad \text{and} \qquad \frac{\Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}\,(\Rightarrow \Box)$$

$(\Box\{A_1, \ldots, A_n\} = \{\Box A_1, \ldots, \Box A_n\})$.

If $\mathcal{S}4 \vdash F$, then there exists a cut-free derivation $\mathcal{T}$ of a sequent $\Rightarrow F$. It suffices now to construct a normal realization r such that $\mathcal{LP} \vdash \bigwedge \Gamma^r \rightarrow \bigvee \Delta^r$ for any sequent $\Gamma \Rightarrow \Delta$ in $\mathcal{T}$. We will also speak about a sequent $\Gamma \Rightarrow \Delta$ being derivable in $\mathcal{LP}$ meaning $\mathcal{LP} \vdash \bigwedge \Gamma \rightarrow \bigvee \Delta$, or, equivalently, $\Gamma \vdash_{\mathcal{LP}} \bigvee \Delta$, or $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$. Note that in a cut-free derivation $\mathcal{T}$ the rules respect polarities, all occurrences of $\Box$ introduced by $(\Rightarrow \Box)$ are positive, and all negative occurrences are introduced by $(\Box \Rightarrow)$ or by weakening. Occurrences of $\Box$ are related if they occur in related formulas of premises and conclusions of rules; we extend this relationship by transitivity. All positive occurrences of $\Box$ in $\mathcal{T}$ are naturally split into disjoint *families* of related ones. We call a family *essential* if it contains at least one case of the $(\Rightarrow \Box)$ rule.

Now the desired r will be constructed by steps 1 – 3 described below. We reserve a large enough set of proof variables as *provisional variables.*

Step 1. For every negative family and non essential positive family we replace all occurrences of $\Box$ by x: for a fresh proof variable x.

Step 2. Pick an essential family f, enumerate all the occurrences of rules $(\Rightarrow \Box)$, which introduce boxes of this family. Let n_f be the total number of such rules for the family f. Replace all boxes of the family f by the term

$$(v_1 + \ldots + v_{n_f}),$$

where v_i's are fresh provisional variables. The resulting tree $\mathcal{T}_0$ is labelled by $\mathcal{LP}$ formulas, since all occurrences of the kind $\Box X$ in $\mathcal{T}$ are replaced by $t{:}X$ for the corresponding t.

Step 3. Replace the provisional variables by proof polynomials as follows. Proceed from the leaves of the tree to its root. By induction on the depth of a node in $\mathcal{T}_0$ we establish that after the process passes a node, a sequent assigned to this node becomes derivable in $\mathcal{LP}$. The axioms $S \Rightarrow S$ and $\bot \Rightarrow$ are derivable in $\mathcal{LP}$. For every rule other than $(\Rightarrow \Box)$ we do not change the realization of formulas, and just establish that the concluding sequent is provable in $\mathcal{LP}$ given that the premises are. Moreover, every move down in the tree $\mathcal{T}_0$ other than $(\Rightarrow \Box)$ is a rule of the system $\mathcal{LPG}$, therefore, the induction steps corresponding to these moves follow easily from the equivalence of $\mathcal{LP}$ and $\mathcal{LPG}$.

Let an occurrence of the rule $(\Rightarrow \Box)$ have number i in the numbering of all rules $(\Rightarrow \Box)$ from a given family f. This rule already has a form

$$\frac{y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow Y}{y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow (u_1 + \ldots + u_{n_f}){:}Y}\,,$$

where $y_1, \ldots, y_k$ are proof variables, $u_1, \ldots, u_{n_f}$ are proof polynomials, and u_i is a provisional variable. By the induction hypothesis, the premise sequent $y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow Y$ is derivable in $\mathcal{LP}$, which yields a derivation of

$$y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow Y.$$

By Lifting (Lemma 1.2), construct a proof polynomial $t(y_1, \ldots, y_n)$ such that

$$y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow t(y_1, \ldots, y_n){:}Y$$

is derivable in $\mathcal{LP}$. Since

$$\mathcal{LP} \vdash t{:}Y \rightarrow (u_1 + \ldots + u_{i-1} + t + u_{i+1} + \ldots + u_{n_f}){:}Y$$

we have

$$\mathcal{LP} \vdash y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow (u_1 + \ldots + u_{i-1} + t + u_{i+1} + \ldots + u_{n_f}){:}Y.$$

Now substitute $t(y_1, \ldots, y_n)$ for u_i everywhere in the tree $\mathcal{T}_0$. Note, that $t(y_1, \ldots, y_n)$ has no provisional variables, there is one provisional variable (namely u_i) less in the entire $\mathcal{T}_0$. All sequents derivable in $\mathcal{LP}$ remain so, the conclusion of the given rule ($\Rightarrow \Box$) became derivable, and the induction step is complete.

Eventually, we substitute terms of non-provisional variables for all provisional variables in $\mathcal{T}_0$ and establish that the corresponding root sequent of $\mathcal{T}_0$ is derivable in $\mathcal{LP}$. Note that the realization of $\Box$'s built by this procedure is normal.

◀

6.3 Corollary. (Arithmetical completeness of $\mathcal{S}4$.) *$\mathcal{S}4 \vdash F$ iff there is a realization r and a constant specification $\mathcal{CS}$ such that F^r is $\mathcal{CS}$-valid.*

6.4 Comment. It follows from 6.1 and 6.2 that $\mathcal{S}4$ is nothing but a lazy version of $\mathcal{LP}$ when we don't keep track on the proof polynomials assigned to the occurrences of $\Box$. Each theorem of $\mathcal{S}4$ admits a decoding via $\mathcal{LP}$ as a statement about specific proofs. The language of $\mathcal{LP}$ is more rich than the one of $\mathcal{S}4$. In particular, $\mathcal{S}4$ theorems admit essentially different realizations in $\mathcal{LP}$. For example, consider two theorems of $\mathcal{LP}$ having the same modal projection: $x{:}F \vee y{:}F \rightarrow (x + y) : F$ and $x : F \vee x : F \rightarrow x : F$. The former of these formulas is a meaningful specification of the operation "+". In a contrast, the latter one is a trivial tautology. So, $\mathcal{LP}$ is the right logic of provability, and $\mathcal{S}4$ should be considered as a lazy higher order language on top of $\mathcal{LP}$. A general recipe for using $\mathcal{S}4$ as a provability logic might be the following: derive in $\mathcal{S}4$ or reason about $\mathcal{S}4$ using a conventional modal logic technique as before, translate the results into $\mathcal{LP}$ to recover their true provability meaning.

Let $\top(F)$ denote a translation of an intuitionistic formula F into the plain modal language which puts the prefix $\Box$ in front of all atoms and implications in F. It is well-known that $\mathcal{I}nt \vdash F$ *iff* $\mathcal{S}4 \vdash \top(F)$ ([12]).

6.5 Corollary. (Realization of intuitionistic logic) *For any $\mathcal{I}nt$-formula F*

$$\mathcal{I}nt \vdash F \quad \Leftrightarrow \quad \mathcal{LP} \vdash (\top(F))^r \text{ for some realization } r.$$

6.6 Comment. One cannot realize even a formula from 1.4 without "+". Moreover, the "+"free fragment of $\mathcal{LP}$ cannot realize modal *modus ponens*: there are modal formulas A and $A \to B$ both realizable without "+" such that B is not realizable without "+". The "+"free fragment of $\mathcal{LP}$ is not complete with respect to the class of all deterministic proof predicates. In order to make it complete one has to add a functionality principle from [3]. The completeness of the resulting system $\mathcal{FLP}$ with respect to deterministic proof systems was established by V. Krupski in ([23]). $\mathcal{FLP}$ does not have a modal counterpart. For example, $\mathcal{FLP}$ derives a principle $\neg(x{:}A \wedge x{:}(A \to A))$, which is false in any normal modal logic. One of the surprises of the logic of proofs is that

provability as a modal operator corresponds to non-deterministic proofs.

It is easy to see that the "+"free fragment of $\mathcal{LP}$ cannot realize the intuitionistic formula $(a \vee b) \to (\neg a \to b)$, if we use the translation $\top$ from [12]. However, this formula becomes realizable if we use the original Gödel translation from [16], defined by the clauses $g(P \to Q) = \Box g(P) \to \Box g(Q)$ and $g(P \vee Q) = \Box g(P) \vee \Box g(Q)$.

6.7 Corollary. (Arithmetical completeness of $\mathcal{I}nt$.) $\mathcal{I}nt \vdash F$ *iff there is a realization r and constant specification $\mathcal{CS}$ such that $\top(F)^r$ $\mathcal{CS}$-valid.*

Kolmogorov's interpretation of intuitionistic logic $\mathcal{I}nt$ as a "*calculus of problems*" ([20]) can be made explicit via $\mathcal{LP}$. Let F be a formula in the language of $\mathcal{I}nt$. Consider the translation $k(F)$ of F into the language of $\mathcal{S}4$ along the lines of the (informal) definitions in ([20]): $k(P) = \Box P$ for a sentence letter P, $k(\bot) = \bot$, $k(A \to B) = \Box[k(A) \to k(B)]$, $k(A \wedge B) = \Box[k(A) \wedge k(B)]$, $k(A \vee B) = \Box[k(A) \vee k(B)]$. It is easy to verify that

$$\mathcal{I}nt \vdash F \quad \Leftrightarrow \quad \mathcal{S}4 \vdash k(F).$$

6.8 Definition. Let F be a formula in the intuitionistic propositional language. A *Kolmogorov realization* $K(F)$ of F is an $\mathcal{LP}$-formula $[k(F)]^r$, where r is a realization of modalities

in $k(F)$ by proof polynomials. A formula F is *Kolmogorov realizable* if it has a Kolmogorov realization $K(F)$ provable in $\mathcal{LP}$. Under a realization K propositional atoms in F become *atomic problems*, namely statements of the sort $t{:}S$ meaning "*t is a proof of S*". Intuitionistic connectives are given precise meaning according to [20].

6.9 Theorem. (Completeness of $\mathcal{I}nt$ with respect to the Kolmogorov realization)

$$\mathcal{I}nt \vdash F \quad \Leftrightarrow \quad F \textit{ is Kolmogorov realizable.}$$

Proof. Either $\mathcal{I}nt \vdash F$, then $\mathcal{S}4 \vdash k(F)$ and, by theorem 6.2, there exists a proof realization r such that $\mathcal{LP} \vdash [k(F)]^r$. Put $K(F) = [k(F)]^r$. Or $\mathcal{S}4 \nvdash k(F)$ and, by lemma 6.1, $\mathcal{LP} \nvdash [k(F)]^r$ for any realization r.
◀

6.10 Comment. By theorem 6.9, the logic of proofs $\mathcal{LP}$ provides a provability semantics for the Kolmogorov "*calculus of problems*". A conjecture, that the "*calculus of problems*" coincides with $\mathcal{I}nt$ was made by Kolmogorov in [20]. Note that such realizability models of $\mathcal{I}nt$ as Kleene realizability ([19]) and Medvedev logic of problems ([29]) provide only necessary conditions for $\mathcal{I}nt$ (cf.[12]). Each of them along with $\mathcal{I}nt$ realize some formulas which are not derivable in $\mathcal{I}nt$.

7 Intuitionistic Logic of Proofs

The Intuitionistic logic of proofs $\mathcal{LP}i$ is defined as a system 1.1 with *A0* being a list of axiom scheme for the propositional intuitionistic logic. $\mathcal{LP}i$ is correct with respect to the natural provability interpretation as a calculus of proofs for either $\mathcal{PA}$ or the intuitionistic arithmetic $\mathcal{HA}$. We do not address the issue of arithmetical completeness of $\mathcal{LP}i$ in this paper.

The Gentzen style system $\mathcal{LPG}i$ for the intuitionistic logic of proofs can be defined as follows (cf. the system **G2i** from [45]). Sequents in $\mathcal{LPG}i$ are all of the form $\Gamma \Rightarrow F$, where Γ is a multiset of $\mathcal{LP}$-formulas, and F is an $\mathcal{LP}$-formula.

Axioms of $\mathcal{LPG}i$ are sequents of the form $P, \Gamma \Rightarrow P$, where P is either a sentence letter or a formula of the sort $t{:}F$, and sequents of the form $\bot, \Gamma \Rightarrow F$.

Rules of $\mathcal{LPG}i$ are

$$\frac{A, B, \Gamma \Rightarrow C}{A \wedge B, \Gamma \Rightarrow C}\,(L\wedge) \qquad\qquad \frac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}\,(R\wedge)$$

$$\frac{A,\Gamma \Rightarrow C \qquad B,\Gamma \Rightarrow C}{A \vee B,\Gamma \Rightarrow C}\,(L\vee) \qquad \frac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_0 \vee A_1}\,(R\vee)\ (i=0,1)$$

$$\frac{\Gamma \Rightarrow A \qquad B,\Gamma \Rightarrow C}{A \to B,\Gamma \Rightarrow C}\,(L\to) \qquad \frac{A,\Gamma \Rightarrow B}{\Gamma \Rightarrow A \to B}\,(R\to)$$

$$\frac{A,A,\Gamma \Rightarrow C}{A,\Gamma \Rightarrow C}\,(LC) \qquad \frac{\Gamma \Rightarrow A \qquad A,\Gamma' \Rightarrow B}{\Gamma,\Gamma' \Rightarrow B}\,(Cut)$$

$$\frac{A,\Gamma \Rightarrow B}{t{:}A,\Gamma \Rightarrow B}\,(L:) \qquad \frac{\Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow !t{:}t{:}A}\,(R!)$$

$$\frac{\Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow \Delta,(t+s){:}A}\,(Rl+) \qquad \frac{\Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow \Delta,(s+t){:}A}\,(Rr+)$$

$$\frac{\Gamma \Rightarrow s{:}(A \to B) \qquad \Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow (s \cdot t){:}B}\,(R\cdot) \qquad \frac{\begin{array}{c}\mathcal{D}\\ \Gamma \Rightarrow \mathbf{A}\end{array}}{\Gamma \Rightarrow c{:}\mathbf{A}}\,(Rc),$$

where $\mathbf{A}$ is an axiom $A0$ - $A4$ of the Hilbert style system for $\mathcal{LP}i$ (definition 1.1.), c is a proof constant and $\mathcal{D}$ is the *standard derivation of* $\Gamma \Rightarrow \mathbf{A}$. Under the standard derivation here we mean the following. If $\mathbf{A}$ is $A0$ (i.e. a propositional axiom), then $\mathcal{D}$ is the straightforward cut-free derivation of $\Gamma \Rightarrow \mathbf{A}$ in the Gentzen style system for $\mathcal{I}nt$. For axioms $A1$ - $A4$ the standard derivations are

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathcal{D}'\\ F,\Gamma \Rightarrow F\end{array}}{t{:}F,\Gamma \Rightarrow F}}{\Gamma \Rightarrow t{:}F \to F}}{},$$

where $\mathcal{D}'$ is the straightforward cut-free derivation of $F,\Gamma \Rightarrow F$ in the Gentzen style system for $\mathcal{I}nt$.

$$\dfrac{\dfrac{s{:}(F \to G),t{:}F,\Gamma \Rightarrow s{:}(F \to G) \qquad s{:}(F \to G),t{:}F,\Gamma \Rightarrow t{:}F}{s{:}(F \to G),t{:}F,\Gamma \Rightarrow (s \cdot t){:}G}}{\Gamma \Rightarrow s{:}(F \to G) \to (t{:}F \to (s \cdot t){:}G)}$$

$$\frac{\dfrac{t{:}F,\Gamma \Rightarrow t{:}F}{t{:}F,\Gamma \Rightarrow\ !t{:}t{:}F}}{\Gamma \Rightarrow t{:}F \to !t{:}t{:}F} \qquad \frac{\dfrac{t{:}F,\Gamma \Rightarrow t{:}F}{t{:}F,\Gamma \Rightarrow (t+s){:}F}}{\Gamma \Rightarrow t{:}F \to (t+s){:}F}$$

Under $\mathcal{LPG}i^-$ we mean a cut-free fragment of $\mathcal{LPG}i$.

7.1 Theorem. *Cut elimination holds for $\mathcal{LP}i$.*

Proof. We shall deliver a syntactical proof that $\mathcal{LPG}i \vdash \Gamma \Rightarrow A$ yields $\mathcal{LPG}i^- \vdash \Gamma \Rightarrow A$.

7.2 Definition. A *level* of a cut is the sum of the depths of the deductions of the premises. The *rank* $rk(A, \mathcal{D})$ of a given occurrence of A in a derivation $\mathcal{D}$ is defined by the following induction on the depth of this occurrence in $\mathcal{D}$. For a term or a formula X by $|X|$ we denote the total number of occurrences of propositional, proof variables and constants, propositional and functional symbols in X. If $X \in \{P, \bot, \Gamma\}$ in a derivation $\mathcal{D}$ consisting of an axiom $P, \Gamma \Rightarrow P$ or $\bot, \Gamma \Rightarrow F$, then $rk(X, \mathcal{D}) = |X|$.

For all the rules of $\mathcal{LPG}i$ ranks of the corresponding occurrences of the side formulas coincide. For the rule(L$\wedge$)

$$rk(A \wedge B, \mathcal{D}) \;=\; rk(A, \mathcal{D}) + rk(B, \mathcal{D}) + 1.$$

Similarly for the rule (R$\wedge$), (L$\vee$) and (R$\rightarrow$).

For $(R\vee)$, case $j = 0$,

$$rk(A_0 \vee A_1, \mathcal{D}) \;=\; rk(A_0, \mathcal{D}) + |A_1| + 1,$$

similarly for $j = 1$.

For $(L \rightarrow)$

$$rk(A \rightarrow B, \mathcal{D}) \;=\; rk(A, \mathcal{D}) + rk(B, \mathcal{D}) + 1.$$

For $(L :)$

$$rk(t{:}A, \mathcal{D}) \;=\; rk(A, \mathcal{D}) + |t|.$$

For $(R!)$

$$rk(!t{:}t{:}A, \mathcal{D}) \;=\; rk(t{:}A, \mathcal{D}) + |!t|.$$

For $(Rl+)$

$$rk((t + s){:}A, \mathcal{D}) \;=\; rk(t{:}A, \mathcal{D}) + |s| + 1.$$

For $(Rr+)$

$$rk((t + s){:}A, \mathcal{D}) \;=\; rk(s{:}A, \mathcal{D}) + |t| + 1.$$

For $(R\cdot)$

$$rk((s \cdot t){:}B, \mathcal{D}) \;=\; rk(s{:}(A \rightarrow B), \mathcal{D}) + rk(t{:}A, \mathcal{D}) + |(s \cdot t){:}B|.$$

For (Rc)

$$rk(c{:}\mathbf{A}, \mathcal{D}) \;=\; rk(\mathbf{A}, \mathcal{D}) + |1|.$$

Note that $rk(\mathbf{A}, \mathcal{D} = |\mathbf{A}|$.

For (LC) the rank of the occurrence of A in the conclusion of the rule is the maximum rank of the indicated occurrences of A in the premise sequent. The *rank of the cut rule*

$$\frac{\Gamma \Rightarrow A \qquad A, \Gamma' \Rightarrow B}{\Gamma, \Gamma' \Rightarrow B}\,(Cut)$$

is the maximum rank of the indicated occurrences of A in the premise sequents. The *cutrank* of the deduction $\mathcal{D}$ is the maximum of the ranks of the cuts occurring in $\mathcal{D}$.

From the definitions it follows easily that

1. $|A| \leq rk(A, \mathcal{D})$ and $rk(A, \mathcal{D}) = |A|$ if $\mathcal{D}$ does not use the rule $(\mathrm{R}\cdot)$.
2. $rk(A, \mathcal{D})$ monotonically increases for the related occurrences of A with the increase of depth.

7.3 Lemma. (Rank- and depth-preserving invertibility of the rule $(\mathrm{R}\!\rightarrow)$). *If $\mathcal{D}$ is a derivation of $\Gamma \Rightarrow A \rightarrow B$, then there is a derivation $\mathcal{D}'$ of $A, \Gamma \Rightarrow B$ such that*

1. *the depth of $\mathcal{D}'$ is not greater, then the depth of $\mathcal{D}$,*
2. *the cutrank of $\mathcal{D}'$ equals to the cutrank of $\mathcal{D}$,*
3. *$rk(F\mathcal{D}') = rk(F, \mathcal{D})$ for all formulas from Γ,*
4. *$rk(A, \mathcal{D}') + rk(B, \mathcal{D}') + 1 = rk(A \rightarrow B, \mathcal{D})$.*

Proof. An induction on the depth of $\mathcal{D}$. The base case corresponds to an axiom. Since $A \rightarrow B$ is neither atomic nor of the form $t{:}F$ the case when $A \rightarrow B$ is a principal formula of an axiom is impossible. If $\mathcal{D}$ is an axiom $\bot, \Delta \Rightarrow A \rightarrow B$, then put $\mathcal{D}'$ to be $\bot, A, \Delta \Rightarrow B$. For the induction step consider two possibilities. If $A \rightarrow B$ is the side formula of the last rule in $\mathcal{D}$, then replace $\Delta \Rightarrow A \rightarrow B$ in the premise(s) of the last rule by $A, \Delta \Rightarrow B$, by IH. If $A \rightarrow B$ is the principal formula of the last rule in $\mathcal{D}$, then the deduction ends with

$$\frac{\begin{array}{c}\mathcal{D}_1 \\ A, \Gamma \Rightarrow B\end{array}}{\Gamma \Rightarrow A \rightarrow B}\,.$$

In this case put $\mathcal{D}'$ to be $\mathcal{D}_1$.

◀

7.4 Lemma. (m-reduction) *Let $\mathcal{D}$ be a cut-free derivation of $\Gamma \Rightarrow t : A$. Then there is a derivation $\mathcal{D}'$ of $\Gamma \Rightarrow A$ such that*

1. *the cutrank of $\mathcal{D}'$ is less then the rank of the indicated occurrence of $t{:}A$ in $\mathcal{D}$,*
2. *$rk(F, \mathcal{D}') = rk(F, \mathcal{D})$ for all formulas from Γ of the end sequent,*

3. $rk(A, \mathcal{D}') < rk(t{:}A, \mathcal{D})$ *for* A *and* $t{:}A$ *being the antecedents of the end sequents of* $\mathcal{D}'$ *and* $\mathcal{D}$ *respectively.*

Proof. Induction on the depth of $\mathcal{D}$. If $\mathcal{D}$ is an axiom $t{:}F, \Delta \Rightarrow t{:}F$, then let $\mathcal{D}'$ be the derivation

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ F, \Gamma \Rightarrow F\end{array}}{t{:}F, \Gamma \Rightarrow F}\ (\mathrm{L}{:}),$$

where $\mathcal{D}_1$ is a standard cut-free derivation of $F, \Gamma \Rightarrow F$. Note, that such a derivation does not use the rule $(R\cdot)$, therefore $rk(X, \mathcal{D}') = |X| = rk(X, \mathcal{D})$ for all formulas from Γ. Similarly, $rk(F, \mathcal{D}') = |F| < rk(t{:}F, \mathcal{D})$.

The induction step. The case when $t{:}A$ is a side formula of the last rule in $\mathcal{D}$ is trivial. Let $t{:}A$ be the principal formula of the last rule $(R!)$ in $\mathcal{D}$, then the deduction ends with

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow\, !t{:}t{:}A}\,.$$

In this case $\mathcal{D}_1$ is a cut-free derivations sutisfying also the requirements 2. and 3. of the lemma.

If the last rule in $\mathcal{D}$ is $(Rl+)$, then the deduction ends with

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (t+s){:}A}\,.$$

By the IH, there exists a derivation $\mathcal{D}_1'$ of $\Gamma \Rightarrow A$ satisfying the lemma's conditions for the derivation $\mathcal{D}_1$. Put $\mathcal{D}'$ to be $\mathcal{D}_1'$. The case $(Rr+)$ can be treated similarly.

If the last rule in $\mathcal{D}$ is $(R\cdot)$, then the deduction ends with

$$\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \Gamma \Rightarrow s{:}(A \to B) & \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (s \cdot t){:}B}\,.$$

By the IH, there exist derivations $\mathcal{D}_1'$ of $\Gamma \Rightarrow A \to B$ and $\mathcal{D}_2'$ of $\Gamma \Rightarrow A$ satisfying the lemma's conditions. Take the derivation $\mathcal{D}_1''$ of $A, \Gamma \Rightarrow B$ from the inversion lemma 7.3 and combine the new derivation $\mathcal{D}_3$

$$\dfrac{\begin{array}{cc}\mathcal{D}_2' & \mathcal{D}_1''\\ \Gamma \Rightarrow A & A, \Gamma \Rightarrow B\end{array}}{\Gamma, \Gamma \Rightarrow B}\,.$$

Using the contraction (LC) we get the desired derivation $\mathcal{D}'$ of $\Gamma \Rightarrow B$. It is easy to check that all the requirements of the lemma are met.

If the last rule in $\mathcal{D}$ is $(R\cdot)$, then $\mathcal{D}$ is

$$\frac{\begin{array}{c}\mathcal{D}_1 \\ \Gamma \Rightarrow \mathbf{A}\end{array}}{\Gamma \Rightarrow c{:}\mathbf{A}}\ .$$

Let $\mathcal{D}'$ be $\mathcal{D}_1$.

◀

Now we return to the proof of theorem 7.1. Our strategy will consist of eliminating the uppermost cuts. In order to save expositions of some well known constructions we will refer to the corresponding steps of the proof of the cut elimination theorem 4.1.2 from [45] when convenient.

7.5 Lemma. *Let $\mathcal{D}$ be a derivation ending in a cut*

$$\frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma \Rightarrow A & A, \Gamma' \Rightarrow B\end{array}}{\Gamma, \Gamma' \Rightarrow B}$$

such that $\mathcal{D}$ contains no other cuts. Then we can transform $\mathcal{D}$ into a derivation $\mathcal{D}'$ of the same sequent $\Gamma, \Gamma' \Rightarrow B$ such that cutrank($\mathcal{D}'$)<cutrank($\mathcal{D}$) $= \max\{rk(A, \mathcal{D}_1), rk(A, \mathcal{D}_2)\}$ without an increase of the ranks of the formulas from Γ, Γ', B.

Proof. An induction on the rank of the cut rule, with a subinduction on its level. There are then three possibilities:

1. at least one of $\mathcal{D}_1$, $\mathcal{D}_2$ is an axiom $P, \Gamma \Rightarrow P$ or $\bot, \Gamma \Rightarrow F$;
2. not 1. and the cutformula is not principal in at least one of the premises;
3. not 1. and the cutformula is principal on both sides.

Case 1. Cut can be eliminated at all by the standard reductions ([45]).

Case 2. We permute the cut upward in a stantard way (cf.[45]) without changing its rank as well as the ranks of all formulas in the end sequent of the derivation, until we find ourselves in situations number 1 or number 3.

Case 3. The cutformula is principal in both premises and neither of the premises an axiom. The induction hypothesis is that the claim of lemma has been shown for all cuts of rank less than $rk(A, \mathcal{D})$ and of rank equal $rk(A, \mathcal{D})$, but level less than the one of the given cut. The rules corresponding to propositional connectives are treated in a usual way (cf.[45]). There is

one additional concern here compared to [45]: we have to make sure that our reductions do not increase the ranks of the side formulas from Γ, Γ'. For example, let us consider the case $(R\rightarrow)$. The original deduction is

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_0\\ \Gamma, A \Rightarrow B\end{array}}{\Gamma \Rightarrow A{\rightarrow}B} \qquad \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \Gamma' \Rightarrow A & \Gamma', B \Rightarrow C\end{array}}{\Gamma', A{\rightarrow}B \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

This is transformed into

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_0\\ \Gamma' \Rightarrow A & \Gamma, A \Rightarrow B\end{array}}{\Gamma', \Gamma \Rightarrow B} \qquad \begin{array}{c}\mathcal{D}_2\\ \Gamma', B \Rightarrow C\end{array}}{\Gamma', \Gamma', \Gamma, \Rightarrow C}\ .$$

We have eliminated the old cut but have got two new cuts instead, both having lower ranks. After a number of contractions in the end sequent we get the desired derivation.

The new cases emerge when the cutformula is of the form $t{:}F$. In all those cases the right premise is just introduced by (L:). So, we distinguish the cases by their left premises.

Case $(R!)$. The derivation is

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow !t{:}t{:}A} \qquad \dfrac{\begin{array}{c}\mathcal{D}_2\\ t{:}A, \Gamma' \Rightarrow C\end{array}}{!t{:}t{:}A, \Gamma \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

This is transformed into a derivation with a lower ranked cut

$$\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \Gamma \Rightarrow t{:}A & t{:}A, \Gamma' \Rightarrow C\end{array}}{\Gamma, \Gamma' \Rightarrow C}\ ,$$

Case (Rc) is treated is a similar way.

Case $(Rl+)$ (and $(Rr+)$ by a similar argument). The derivation

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (t+s){:}A} \qquad \dfrac{\begin{array}{c}\mathcal{D}_2\\ A, \Gamma' \Rightarrow C\end{array}}{(t+s){:}A, \Gamma \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

should be transformed into one with a lower cutrank

$$\dfrac{\dfrac{\mathcal{D}_1}{\Gamma \Rightarrow t{:}A} \qquad \dfrac{\mathcal{D}_2}{A, \Gamma' \Rightarrow C}}{t{:}A, \Gamma \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

Case $(R\cdot)$:

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_0}{\Gamma \Rightarrow s{:}(A \to B)} \qquad \dfrac{\mathcal{D}_1}{\Gamma \Rightarrow t{:}A}}{\Gamma \Rightarrow (s \cdot t){:}B} \qquad \dfrac{\dfrac{\mathcal{D}_2}{B, \Gamma' \Rightarrow C}}{(s \cdot t){:}B, \Gamma' \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

We transform it into a derivation with a lower cutrank as follows. By m-reduction (lemma 7.4) without a rank increase we get derivations

$$\dfrac{\mathcal{D}_0'}{\Gamma \Rightarrow A \to B} \qquad \dfrac{\mathcal{D}_1'}{\Gamma \Rightarrow A}$$

From $\mathcal{D}_0'$ by 7.3 without the rank increase of the side formulas we get a derivation

$$\dfrac{\mathcal{D}_0''}{\Gamma, A \Rightarrow B}$$

where

$$rk(A, \mathcal{D}_0''),\ rk(B, \mathcal{D}_0'') < rk(A \to B, \mathcal{D}_0') < rk(s{:}(A \to B), \mathcal{D}_0) < rk((s \cdot t){:}B, \mathcal{D}).$$

The transformed derivation in this case will be

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_1'}{\Gamma \Rightarrow A} \qquad \dfrac{\mathcal{D}_0''}{A, \Gamma \Rightarrow B}}{\Gamma, \Gamma \Rightarrow B} \qquad \dfrac{\mathcal{D}_2}{B, \Gamma' \Rightarrow C}}{\Gamma, \Gamma, \Gamma' \Rightarrow C}$$

Again, use some contractions in the end sequent to get the desired derivation. We have eliminated the old cut and have created two new ones and, may be, some more in $\mathcal{D}_1'$ and $\mathcal{D}_0''$ as a result of the m-reduction. By 7.3 and 7.4 all new cuts have lower rank.

This ends the proof of lemma 7.5.

◀

7.6 Lemma. *Let a derivation $\mathcal{D}$ contains not more than one use of the cut rule. Then by a finite chain of reductions it can be transformed into a cut-free derivation $\mathcal{D}'$ of the same end sequent without changing the ranks of the formulas from the end sequent.*

Proof. An induction on the $n = cutrank(\mathcal{D})$. The base case $n = 0$. Then $\mathcal{D}$ is already cut-free. The induction step. Assume $n > 0$, thus, $\mathcal{D}$ contains a cut. Without loss of generality assume that the cute rule is the last rule in $\mathcal{D}$. By Lemma 7.5 transform $\mathcal{D}$ into $\mathcal{D}_1$ with $cutrank(\mathcal{D}_1) < cutrank(\mathcal{D})$. Beginning with the uppermost cuts in $\mathcal{D}_1$ eliminate them all by the IH.
◀

To conclude the proof of Theorem 7.1 use Lemma 7.6 to eliminate every cut in a given derivation beginning with the uppermost ones.
◀

8 Natural deduction system and λ-terms for Logic of Proofs

8.1 Definition. The *natural deduction system $\mathcal{LPN}$ for $\mathcal{LP}$* is obtained from a usual natural deduction system for propositional logic (cf. [13], [45]) in the language of $\mathcal{LP}$ extended by the following rules

$$\frac{s:(A \to B) \quad t:A}{(s \cdot t):B}\,(\cdot I) \qquad \frac{t:A}{A}\,(:E) \qquad \frac{t:A}{!t:t:A}\,(!I)$$

$$\frac{t:A}{(t+s):A}\,(+I) \qquad \frac{t:A}{(s+t):A}\,(+I) \qquad \frac{\begin{array}{c}\mathcal{D}\\ \mathbf{A}\end{array}}{c:\mathbf{A}}\,(cI)$$

where **A** is an axiom of the Hilbert version of $\mathcal{LP}$ (Section 1), c is a proof constant, and $\mathcal{D}$ is the *standard derivation* of **A**. Under the standard derivation of **A** we mean the following. If **A** is an axiom $A0$, then $\mathcal{D}$ is the straightforward normal derivation of **A** in the natural deduction system for $\mathcal{I}nt$. For other axioms $A1 - A4$ the standard derivations are

$$\frac{\dfrac{[t:A]}{A}}{t:A \to A} \qquad \frac{\dfrac{\dfrac{[s:(A \to B)] \quad [t:A]}{(s \cdot t):B}}{t:A \to (s \cdot t):B}}{s:(A \to B) \to (t:A \to (s \cdot t):B)}$$

$$\dfrac{\dfrac{[t:A]}{!t:t:A}}{t:A\to !t:t:A} \qquad\qquad \dfrac{\dfrac{[t:A]}{(t+s):A}}{t:A\to (t+s):A}$$

Note that a standard derivation has no undischarged premises. Under $\mathcal{LPN} \vdash \Gamma \Rightarrow A$ we mean "*A is derivable in $\mathcal{LPN}$ from assumptions* Γ".

A standard theorem relating Hilbert, Gentzen and natural style derivations in $\mathcal{LP}$ holds. Namely the following are equivalent

1. $\Gamma \vdash_{\mathcal{LP}} A$
2. $\mathcal{LPG} \vdash \Gamma \Rightarrow A$
3. $\mathcal{LPN} \vdash \Gamma \Rightarrow A$.

This fact is established by the standard mutual simulations of derivations in all three systems (cf. section 3.3 in [45]). In fact for any source derivation of size s the simulation runs in polynomial time and produces a derivation of size $O(s)$.

The following analog of the Lifting lemma 1.2 holds for $\mathcal{LPN}$.

8.2 Corollary. *If* $\vdash \vec{s}:\Gamma, \Delta \Rightarrow A$, *then for any proof variables* $\vec{y}$ *one can construct a proof polynomial* $t(\vec{x},\vec{y})$ *such that* $\mathcal{LPN} \vdash \vec{s}:\Gamma, \vec{y}:\Delta \Rightarrow t(\vec{s},\vec{y}):A$. *Moreover, if* $\Gamma = \emptyset$ *then the derivation of* $\vec{y}:\Delta \Rightarrow t(\vec{y}):A$ *contains no rules other than* $(\cdot I)$, (cI) *or* $(c^2 I)$.

Proof. Induction of the depth of a derivation. A straightforwrd natural deduction version of Lemma 1.2. The number of steps in the algorithm constructing $\mathcal{D}'$ is bounded by a polynomial of the length of $\mathcal{D}$.
◀

8.3 Definition. *Contractions* for $\mathcal{LPN}$ include all usual contractions for propositional logic ($\wedge-, \vee-, \to$-*contractions, permutation contractions*) (cf. section 6.1.3 in [45]), and the new contractions

$\cdot$-*contraction*

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1 & \mathcal{D}_2\\ s:(A\to B) & t:B\end{matrix}}{(s\cdot t):B}}{B} \qquad \textit{transforms into} \qquad \dfrac{\dfrac{\mathcal{D}_1}{s:(A\to B)} \quad \dfrac{\mathcal{D}_2}{t:A}}{B},$$

+-contraction

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ t:A\end{array}}{(t+s):A}}{A} \qquad \textit{transforms into} \qquad \dfrac{\begin{array}{c}\mathcal{D}\\ t:A\end{array}}{A},$$

!-contraction

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ t:A\end{array}}{!t:t:A}}{t:A} \qquad \textit{transforms into} \qquad \begin{array}{c}\mathcal{D}\\ t:A,\end{array}$$

c-contraction

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \mathbf{A}\end{array}}{c:\mathbf{A}}}{\mathbf{A}} \qquad \textit{transforms into} \qquad \begin{array}{c}\mathcal{D}\\ \mathbf{A}.\end{array}$$

An obvious new permutational contraction should also be added that allow "pulling" the $(:E)$ upwards through the $(\vee E)$ rule. An $\mathcal{LPN}$-derivation $\mathcal{D}$ is normal if no contraction is possible anywhere in $\mathcal{D}$.

8.4 Theorem. *Normalization holds for $\mathcal{LPN}$*

Proof. Given a derivation $\mathcal{D}$ in $\mathcal{LPN}$ of the type $\Gamma \Rightarrow A$ one can construct a normal derivation $\mathcal{D}'$ of the same type $\Gamma \Rightarrow A$. Indeed, for a given derivation of $\Gamma \Rightarrow A$ in $\mathcal{LPN}$ one can construct an $\mathcal{LPG}$-derivation of a sequent $\Gamma \Rightarrow A$. By 5.11, there exists a cut-free $\mathcal{LPG}$-derivation of $\Gamma \Rightarrow A$. A usual transformation of a cut-free $\mathcal{LPG}$-derivation into an $\mathcal{LPN}$-derivation produces a normal $\mathcal{LPN}$-derivation (cf. section 6.3 in [45]). Since both of the transitions from $\mathcal{LPN}$ derivations and back are computable (and feasible), the entire efficiency of the process of proof normalization in $\mathcal{LPN}$ in this proof is determined by an efficiency of normalization of proofs in $\mathcal{LPG}$. An easy adaptation of a constructive normalization theorem for $\mathcal{LPN}i$ independently establishes a direct algorithm of normalizing derivations in $\mathcal{LPN}$ as well.

◀

We leave the description of a direct realization algorithm for the normal deduction system for $\mathcal{S}4$ in $\mathcal{LPN}$ based on 6.2 as a useful exercise.

8.5 Definition. Under $\mathcal{LPN}i$ we mean an intuitionistic version of of $\mathcal{LPN}$ which is obtained from $\mathcal{LPN}$ by omitting the double negation rule

$$\frac{\begin{array}{c}[\neg A]\\ \mathcal{D}\\ \bot\end{array}}{A}$$

8.6 Theorem. (From Gentzen to normal deductions in the intuitionistic $\mathcal{LP}$)

$$\mathcal{LPG}i \vdash \Gamma \Rightarrow A \quad \textit{if and only if} \quad \mathcal{LPN}i \vdash \Gamma \Rightarrow A.$$

Moreover, a cut-free derivation in $\mathcal{LPG}i$ transforms into a normal derivation in $\mathcal{LPN}i$.

Proof. A usual argument in the style of Section 6.3 from [45].
◀

8.7 Corollary. *Normalization holds for $\mathcal{LPN}i$*

Proof. Take a derivation of the sort $\Gamma \Rightarrow A$ in $\mathcal{LPN}$, transform it into a derivation of $\Gamma \Rightarrow A$ in $\mathcal{LPG}i$, perform a cut elimination, and transform the resulting cut-free proof back into an $\mathcal{LPN}i$ derivation. By Theorem 8.6, the resulting derivation is normal. As an exercise one could write down a direct algorithm of normalization of derivations in $\mathcal{LPN}i$ that will essentially repeat the reduction steps for cut elimination in $\mathcal{LPG}i$. Moreover, on the basis of the reductions from the proof of Theorem 7.1 one could establish a strong normalization property of $\mathcal{LPG}i$ and $\mathcal{LPN}i$.
◀

Extending the term calculus for the intuitionistic logic ([45]) we can identify the full $\mathcal{LPN}i$ with a system of typed λ-terms in a natural way.

8.8 Definition. We define a λ-term calculus $\mathcal{LPN}i\lambda$ for the full $\mathcal{LPN}i$. The language of $\mathcal{LPN}i\lambda$ has only formulas of the type $t{:}F$ where F is an $\mathcal{LP}$-formula, and t is a term built from the proof variables and proof constants by atomic operations $\mathbf{p}$, $\mathbf{p}_j$, $\mathbf{k}_j$, $E^{\vee}_{u,v}$, $E^{\bot}_{A}$, App, $\mathbf{P}$, $\mathbf{U}$, $\mathbf{B}$, $\mathbf{S}_j$, $\mathbf{C}$, (j=0,1), and λ-abstraction. The arities of the operations will be made clear in the rules. The first eight clauses of come directly from the term calculus for $\mathcal{I}nt$ ([45], 2.2.2).

We will omit an obvious description of free and bounded variables. As usual, $[A]$ denotes a discharged premise A. In the derivations denoted in this definition by

$$\begin{array}{c}[w:F]\\ \vdots\\ p:G\end{array}$$

the variable w occurs free neither in F, G nor in any undischarged premise of the derivation.

$$y:F \qquad \frac{t:\bot}{E^{\bot}_{A}(t):A}\,(\bot Et)$$

$$\frac{s:A \quad t:B}{\mathbf{p}(s,t):(A\wedge B)}\,(\wedge It) \qquad \frac{t:(A_0\wedge A_1)}{\mathbf{p_j}(t):A_j}\,j\in\{0,1\},\,(\wedge Et)$$

$$\frac{t:A_j}{\mathbf{k}_j(t):(A_0\vee A_1)}\,j\in\{0,1\},\,(\vee It) \qquad \frac{t:(A\vee B) \quad \begin{array}{c}[u:A]\\ \vdots\\ s:C\end{array} \quad \begin{array}{c}[v:B]\\ \vdots\\ s':C\end{array}}{E^{\vee}_{u,v}(t,s,s'):C}\,(\vee Et)$$

$$\frac{\begin{array}{c}[u:A]\\ \vdots\\ t:B\end{array}}{\lambda u.t:(A\to B)}\,(\to It) \qquad \frac{s:(A\to B) \qquad t:A}{App(s,t):B}\,(\to Et)$$

$$\frac{q:s:(A\to B) \qquad r:t:A}{\mathbf{P}(q,r):(s\cdot t):B}\,(\cdot It) \qquad \frac{q:t:A}{\mathbf{U}(q):A}\,(:Et)$$

$$\frac{q:t:A}{\mathbf{B}(q):!t:t:A}\,(!tI) \qquad \frac{q:t_j:A}{\mathbf{S}_j(q):(t_0+t_1):A}\,j\in\{0,1\},\;(+I)$$

Note that the list of rules above suffices to build a λ-term p without free variables which internalize in $\mathcal{LPN}i\lambda$ the standard $\mathcal{LPN}i$-derivation of an axiom $\mathbf{A}$. In particular, $\mathcal{LPN}i\lambda \vdash p: \mathbf{A}$. For example, the λ-term version of the standard derivation of axiom $A3$ is

$$\dfrac{\dfrac{v:t:F}{\mathbf{B}(v)\ !t:t:F}}{\lambda v.\mathbf{B}(v):(t:F\to !t:t:F)}$$

The last rule of $\mathcal{LPN}i\lambda$ is

$$\dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ p:\mathbf{A}\end{array}\ (cIt),}{\mathbf{C}(p):c:\mathbf{A}}$$

where $\widetilde{\mathcal{D}}$ is the λ-term version of the standard derivation of $\mathbf{A}$ in $\mathcal{LPN}i$.

8.9 Definition. In the term notation the *contractions* for the $\mathcal{LPN}i\lambda$ are

1. $\mathbf{p}_j(\mathbf{p}(t_0,t_1))$ *cont* t_j $(j\in\{0,1\})$,
2. $E^{\vee}_{x_0,x_1}(\mathbf{k}_j t,t_0,t_1)$ *cont* $t_j[x/t]$,
3. $App(\lambda x.t,s)$ *cont* $t[x/s]$,
4. $\mathbf{U}(\mathbf{B}(t))$ *cont* t,
5. $\mathbf{U}(\mathbf{C}(t))$ *cont* t,
6. $\mathbf{U}(\mathbf{P}(t_0,t_1))$ *cont* $App(\mathbf{U}(t_0),\mathbf{U}(t_1))$,
7. $\mathbf{U}(\mathbf{S}_j(t))$ *cont* $\mathbf{U}(t)$,
8. $f[E^{\vee}_{x_0,x_1}(\mathbf{k}_j t,t_0,t_1)]$ *cont* $E^{\vee}_{x_0,x_1}(\mathbf{k}_j t,f[t_0],f[t_1])$, where f is another eliminating operator (i.e. one of $\mathbf{p}_j$, *App*, $\mathbf{U}$).

The contractions 1 - 5 are the called *detour contractions*, 6 - 8 are *permutation contractions*. A λ-term t is normal if no contractions are possible in t.

It follows from the definitions that

$$\mathcal{LPN}i\vdash \Gamma\Rightarrow A \quad \textit{iff} \quad \mathcal{LPN}i\lambda\vdash \vec{x}:\Gamma\Rightarrow t(\vec{x}):A \quad \textit{for some } \mathcal{LPN}i\lambda\textit{-term } t.$$

8.10 Theorem. (Normalization of $\mathcal{LPN}i\lambda$-terms) *Every λ-term for $\mathcal{LPN}i\lambda$ is normalizing.*

Proof. Translate the proof of theorem 8.4 from the language of derivations into the language of λ-terms. In fact one can establish a strong normalization of $\mathcal{LPN}i\lambda$-terms with respect to the contractions 8.9.
◀

9 Abstraction in Logic of Proofs

In this section we show that $\mathcal{LP}$ provides a standard provability semantics for the operator of λ-abstraction. This matches our earlier observation (Section 6), that $\mathcal{S}4$-modality is realized by proof polynomials. Thus modality and λ-terms are objects of the same sort, namely, they are all proof polynomials. Through a realization in $\mathcal{LP}$ both modality and modal lambda terms receive a uniform provability semantics.

The defined abstraction operator $\lambda^* x$ on proof polynomials below is a natural extension of the defined λ-abstraction operator $\lambda^* x$ in combinatory logic (cf. [45]).

9.1 Definition. An $\mathcal{LPN}i$-derivation $\mathcal{D}$ is *pure* if it uses no rules other than $(\cdot I)$, (cI), and a *closed version of* $(!I)$ where the principal proof polynomial t contains no variables. It is clear that every pure derivation is normal since it has no elimination rules.

9.2 Lemma. (Definable abstraction) *Let $\mathcal{D}$ be a pure $\mathcal{LPN}i$-derivation of a type*

$$\vec{p}{:}\Gamma, x{:}A \Rightarrow t(x){:}B$$

such that x does not occur in $\vec{p}{:}\Gamma, A, B$. Then one may construct a proof polynomial $\lambda^* x.t(x)$ and a pure $\mathcal{LPN}i$-derivation $\mathcal{D}'$ of the type

$$\vec{p}{:}\Gamma \Rightarrow \lambda^* x.t(x){:}(A \to B).$$

Proof. The base case is the depth of $\mathcal{D}$ equals one. There are two possibilities.

1. $\mathcal{D}$ is $t(x){:}B$ and $t(x)$ contains an occurrence of x. Then $t(x){:}B = x{:}A$. Indeed, by the definition of a natural derivation of the depth 1, the formula $t(x){:}B$ should occur in $\vec{p}{:}\Gamma, x{:}A$. Since x does not occur in $\vec{p}{:}\Gamma, A, B$ the only remaining possibility is when $t(x){:}B$ coincides with $x{:}A$. Let $\mathcal{D}'$ be the pure derivation (without undischarged premises) of $(a \cdot b \cdot c){:}(A \to A)$ where a, b, c are proof constants specified by the conditions (cf. [45], section 1.3.6.)

$a{:}([A \to ((A \to A) \to A)] \to [(A \to (A \to A)) \to (A \to A)])$
$b{:}[A \to ((A \to A) \to A)]$

$c{:}[A{\to}(A{\to}A)]$.

Let $\lambda^*x.x = (a \cdot b \cdot c)$.

2. $\mathcal{D}$ is $t{:}B$ and t does not contain an occurrence of x. Then $t{:}B \in \vec{p}{:}\Gamma$. Let $\mathcal{D}'$ be

$$\dfrac{\dfrac{\dfrac{\dfrac{[B]}{A{\to}B}}{B{\to}(A{\to}B)}\,(cI)}{b{:}(B{\to}(A{\to}B))} \qquad t{:}B}{(b \cdot t){:}(A{\to}B)}\,(\cdot I)\,.$$

Let $\lambda^*x.t = b \cdot t$.

The induction step corresponding to the closed $(!I)$ rule is treated similarly to the case 2. Consider the case $(\cdot I)$. Let a derivation $\mathcal{D}$ from the premises $\vec{p}{:}\Gamma, x{:}A$ end with

$$\dfrac{s{:}(Y{\to}B) \qquad t{:}Y}{(s \cdot t){:}B}\,.$$

By the IH, we have already built pure derivations from the premises $\vec{p}{:}\Gamma$ of $\lambda^*x.s{:}(A{\to}(Y{\to}B))$ and $\lambda^*x.t{:}(A{\to}Y)$. From them we construct a pure derivation $\mathcal{D}'$

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{(A{\to}(Y{\to}B)){\to}((A{\to}Y){\to}(A{\to}B))}}{c{:}((A{\to}(Y{\to}B)){\to}((A{\to}Y){\to}(A{\to}B)))} \quad \lambda^*x.s{:}(A{\to}(Y{\to}B))}{(c \cdot \lambda^*x.s){:}((A{\to}Y){\to}(A{\to}B))} \qquad \lambda^*x.t{:}(A{\to}Y)}{(c \cdot \lambda^*x.s \cdot \lambda^*x.t){:}(A{\to}B)}\,,$$

where $\mathcal{D}_1$ is the standard derivation of a propositional axiom. Let $\lambda^*x.(s{\cdot}t) = (c{\cdot}\lambda^*x.s{\cdot}\lambda^*x.t)$. ◀

9.3 Comment. In $\mathcal{LPN}i$ λ-abstraction is presented by a set of proof polynomials depending on a context (e.g. an $\mathcal{LPN}i$-derivation). In this respect the realization from 9.2 of λ-abstraction by proof polynomials is similar the realization of $\mathcal{S}4$-modality which is decomposed in 6.2 into a set of proof polynomials depending on a context (there an $\mathcal{S}4$-derivation).

The operation λ^* suffices to emulate the traditional λ-abstraction. In fact it cannot be easily extended from the pure to more general derivations without sacrificing some desired properties. We need to keep the format $\vec{p}{:}\Gamma, x{:}A \Rightarrow t(x){:}B$ throughout all the derivation $\mathcal{D}$ in order to preserve an inductive character of the definition. The restriction "x does not

occur in $\vec{p}{:}\Gamma, A, B$" is needed to guarantee the correctness of β-conversion for λ^*-abstraction (below), though it rules out $(!I)$. Note, that the rule $(!I)$ does not admit abstraction anyway. Indeed, in $\mathcal{LPN}$ we have

$$x{:}A \Rightarrow !x{:}x{:}A,$$

but for no proof polynomial p

$$\Rightarrow p{:}(A{\to}x{:}A)$$

since $A{\to}x{:}A$ is not provable in $\mathcal{LP}$.

9.4 Definition. The dual operation to λ-abstraction is *β-conversion*

$$(\lambda x^A.t^B)s^A \quad \longrightarrow_\beta \quad t^B[x^A/s^A].$$

β-conversion is naturally presented as the following transformation of pure derivations in $\mathcal{LPN}i$:

$$\dfrac{\begin{array}{c}[x{:}A]\\ \vdots\\ t(x){:}B\end{array}}{\lambda^* x t(x){:}(A{\to}B)} \quad \begin{array}{c}\mathcal{D}\\ s{:}A\end{array} \qquad \textit{transforms into} \qquad \begin{array}{c}\mathcal{D}\\ s{:}A\\ \vdots\\ t(s){:}B,\end{array}$$

$$(\lambda^* x t(x) \cdot s){:}B$$

The rule of *η-conversion*

$$(\lambda x^A.t^B)s^A \quad \longrightarrow_\eta \quad t \qquad \textit{if } x \textit{ is not free in } t$$

is treated in the same way. Finally, *α-conversion* corresponds to an obviously valid rule of renaming bounded variables in $\mathcal{LPN}i$-derivations with abstraction.

All other standard λ-term constructors for $\mathcal{I}nt$ can also be realized as operations on proof polynomials. This is a straightforward corollary of the fact that $\mathcal{I}nt$ is a fragment of $\mathcal{LPN}i$ and of the Lifting rule for $\mathcal{LPN}i$. Indeed, if $\mathcal{LPN}i \vdash \Gamma \Rightarrow B$, then by induction on the given proof one can construct a proof polynomial $p(\vec{y})$ such that $\mathcal{LPN} \vdash \vec{y}{:}\Gamma \Rightarrow p(\vec{y}){:}B$. However, for the sake of clear presentation of λ-terms as proof polynomials we will explicitly build the proof polynomials corresponding to standard λ-terms constructors.

9.5 Definition. We define a list of *standard translations* of term constructors from $\mathcal{LPN}i\lambda$ to corresponding derivations in $\mathcal{LPN}i$.

λ-term constructor	*corresponding derivation in $\mathcal{LPN}i$*

$$\frac{\begin{array}{c}y{:}F\\ \vdots\\ s{:}A \quad t{:}B\end{array}}{\mathbf{p}(s,t){:}(A\wedge B)} \qquad \frac{\begin{array}{c}y{:}F\\ \vdots\\ \dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c{:}(A\to(B\to A\wedge B))\end{array} \qquad s{:}A}{(c\cdot s){:}(B\to A\wedge B)} \quad t{:}B\end{array}}{(c\cdot s\cdot t){:}(A\wedge B)}$$

$$\frac{t{:}(A_0\wedge A_1)}{\mathbf{p_j}(t){:}A_j}\ j\in\{0,1\} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c{:}(A_0\wedge A_1\to A_j)\end{array} \qquad t{:}(A_0\wedge A_1)}{(c\cdot t){:}A_j}$$

$$\frac{t{:}A_j}{\mathbf{k}_j(t){:}(A_0\vee A_1)}\ j\in\{0,1\} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c{:}(A_j\to A_0\vee A_1)\end{array} \qquad t{:}A_j}{(c\cdot t){:}(A_0\vee A_1)}$$

$$\frac{t{:}(A\vee B) \quad \begin{array}{c}[u{:}A]\\ \vdots\\ s{:}C\end{array} \quad \begin{array}{c}[v{:}B]\\ \vdots\\ s'{:}C\end{array}}{E^{\vee}_{u,v}(t,s,s'){:}C} \qquad \frac{\begin{array}{c}\mathcal{D}_1 \qquad \mathcal{D}_2\\ \dfrac{(c\cdot\lambda^*u.s){:}((B\to C)\to(A\vee B\to C))}{(c\cdot\lambda^*u.s\cdot\lambda^*v.s'){:}(A\vee B\to C)} \quad \mathcal{D}_3\end{array} \qquad t{:}(A\vee B)}{(c\cdot\lambda^*u.s\cdot\lambda^*v.s'\cdot t){:}C}$$

where $\mathcal{D}_1$ is

$$\begin{array}{c}\widetilde{\mathcal{D}}\\ c{:}((A\to C)\to((B\to C)\to(A\vee B\to C)))\end{array}$$

$\mathcal{D}_2$ and $\mathcal{D}_3$ are

$$\frac{\begin{array}{c}[u{:}A]\\ \vdots\\ s{:}C\end{array}}{\lambda^*u.s{:}(A\to C)} \qquad \frac{\begin{array}{c}[v{:}B]\\ \vdots\\ s'{:}C\end{array}}{\lambda^*v.s'{:}(B\to C)}$$

$$\frac{\begin{array}{c}[u:A]\\ \vdots\\ t:B\end{array}}{\lambda u.t:(A\to B)} \qquad \frac{\begin{array}{c}[u:A]\\ \vdots\\ t:B\end{array}}{\lambda^* u.t:(A\to B)}$$

$$\frac{s:(A\to B) \qquad t:A}{App(s,t):B} \qquad \frac{s:(A\to B) \qquad t:A}{(s\cdot t):B}$$

$$\frac{t:\bot}{E^{\bot}_{A}(t):A} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c:(\bot\to A) \qquad t:\bot\end{array}}{(s\cdot t):A}$$

$$\frac{u:s:(A\to B) \quad v:t:A}{\mathbf{P}(u,v):(s\cdot t):B} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c:(s:(A\to B)\to(t:A\to(s\cdot t):B)) \qquad u:s:(A\to B)\end{array}}{(c\cdot u):(t:A\to(s\cdot t):B)} \quad v:t:A}{(c\cdot u\cdot v):(s\cdot t):B}$$

$$\frac{v:t:A}{\mathbf{U}(v):A} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c:(t:A\to A) \qquad v:t:A\end{array}}{(c\cdot v):A}$$

$$\frac{v:t:A}{\mathbf{B}(v):!t:t:A} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c:(t:A\to !t:t:A) \qquad v:t:A\end{array}}{(c\cdot v):!t:t:A}$$

$$\frac{v:t_j:A}{\mathbf{S}_j(v):(t_0+t_1):A}\ j\in\{0,1\} \qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c:(t_j:A\to(t_0+t_1):A) \qquad v:t_j:A\end{array}}{(c\cdot v):(t_0+t_1):A}$$

$$\dfrac{\dfrac{\mathcal{D}}{p:\mathbf{A}}\ (cIt)}{\mathbf{C}(p):c:\mathbf{A}} \qquad\qquad \dfrac{\dfrac{d:(c:\mathbf{A}\to(\mathbf{A}\to c:\mathbf{A}))}{(d{\cdot}!c):(\mathbf{A}\to c:\mathbf{A})} \qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ c:\mathbf{A}\end{array}}{!c:c:\mathbf{A}} \qquad\qquad t:\mathbf{A}}{(d\cdot\ !c\cdot t):\mathbf{A}}$$

In each case $\widetilde{\mathcal{D}}$ denotes a corresponding standard $\mathcal{LPN}i$-derivation with the end rule (cI).

9.6 Theorem. (Realization of $\mathcal{LPN}i\lambda$ into $\mathcal{LPN}i$) *Under standard translations from 9.5 an $\mathcal{LPN}i\lambda$-derivation $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$ becomes a pure derivation $\vec{x}:\Gamma^r \Rightarrow t^r(\vec{x}):A^r$ in the $\mathcal{LPN}i$.*

Proof. A straightforward induction on an $\mathcal{LPN}i\lambda$-derivation $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$. It is immediate from definition 9.5 and lemma 9.2 that each standard transformation returns a pure derivation.
◀

9.7 Corollary. (Realization of λ-calculus for $\mathcal{I}nt$ into $\mathcal{LPN}i$) *Let $\mathcal{D}$ be a λ-term derivation of the type $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$ in the term calculus for $\mathcal{I}nt$. Standard translations define an effective step by step realization "r" of $\mathcal{D}$ as a derivation $\mathcal{D}'$ of $\vec{x}:\Gamma \Rightarrow t^r(\vec{x}):A$ in the $\mathcal{LPN}i$.*

9.8 Comment. As it is easy to see that $\mathcal{LPN}i\lambda$ (as well as λ-calculus for $\mathcal{I}nt$) can be realized in a small fragment of $\mathcal{LPN}i$ consisting of pure derivations only.

We already have enough ingredients to demonstrate that the Logic of Proofs can emulate not only classical modal logic, but also modal λ-calculi.

Under $\mathbf{IS4}_\Box$ we mean the intuitionistic modal logic on the basis of $\mathcal{S}4$, introduced in [7] (cf. also [26], [38]). An inspection of the proof of theorem 6.2 (realization of modal logic), shows that this theorem holds also for $\mathcal{LP}i$ instead of $\mathcal{LP}$ and $\mathbf{IS4}_\Box$ instead of $\mathcal{S}4$. In other words, the intuitionistic logic of proofs is an explicit version of $\mathbf{IS4}_\Box$ in the same sense that $\mathcal{LP}$ is an explicit version of $\mathcal{S}4$. We will show how $\mathcal{LPG}i$ naturally emulates the modal λ-calculus for $\mathbf{IS4}_\Box$ and thus supplies modal λ-terms with standard provability semantics.

9.9 Theorem. (Realization of modal λ-calculus). *There is an effective step by step realization "r" of any derivation $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$ in the λ-term calculus for $\mathbf{IS4}_\Box$ as a derivation of $\vec{x}:\Gamma^r \Rightarrow t^r(\vec{x}):A^r$ in $\mathcal{LPN}i$.*

Proof. As above all the usual steps of λ-terms formation can be emulated in the Logic of Proofs (here in $\mathcal{LPN}i$). The entire term assignment system for $\mathbf{IS4}_\Box$ is obtained from the usual one for intuitionistic logic by adding two new rules that correspond to "modal" operations on λ-terms "box" and "unbox":

$$\frac{\Delta \Rightarrow s_1{:}\Box A_1 \ \ldots\ \Delta \Rightarrow s_k{:}\Box A_k \quad x_1{:}\Box A_1, \ldots, x_k{:}\Box A_k \Rightarrow t(\vec{x}){:}B}{\Delta \Rightarrow \mathtt{box}(t(\vec{s})){:}\Box B}\ (\Box I)$$

and

$$\frac{\Gamma \Rightarrow t{:}\Box A}{\Gamma \Rightarrow \mathtt{unbox}(t){:}A}\ (\Box E)$$

Let $\vec{p}{:}\,\Gamma \Rightarrow t : A$ be a modal λ term, and let $\mathcal{D}$ be a natural derivation of A from the hypothesis Γ, which is represented by this λ-term. The construction of the desired realization r takes two rounds. First, we realize all the occurrences of $\Box$ in the derivation $\mathcal{D}$ of B from $A_1, \ldots, A_n$ by proof polynomials according to the algorithm from 6.2. As a result, we get a realization $*$ of modalities in $\mathcal{D}$ such that $\mathcal{LPN}i \vdash \Sigma^* \Rightarrow F^*$ holds for every intermediate derivation $\Sigma \Rightarrow F$ in $\mathcal{D}$. The second round produces the desired realization r and proceeds by an induction on the steps of the λ-term construction (i.e. on the construction of $\mathcal{D}$). Without loss of generality we assume, that the proof variables used in the first round for $*$ are all different from the ones we use in the second round.

At the nodes of $\mathcal{D}$ corresponding to intuitionistic connectives use standard translations from 9.5. At a $(\Box I)$ node perform a natural deduction step is performed:

$$\frac{\Delta \Rightarrow \Box A_1 \ \ldots\ \Delta \Rightarrow \Box A_k \qquad \Box A_1, \ldots, \Box A_k \Rightarrow B}{\Delta \Rightarrow \Box B}.$$

The corresponding step of the modal λ-term assigning process is

$$\frac{\vec{u}{:}\Delta \Rightarrow s_1(\vec{u}){:}\Box A_1 \ \ldots\ \vec{u}{:}\Delta \Rightarrow s_k(\vec{u}){:}\Box A_k \qquad x_1{:}\Box A_1, \ldots, x_k{:}\Box A_k \Rightarrow t(\vec{x}){:}B}{\vec{u}{:}\Delta \Rightarrow \mathtt{box}(t)(\vec{s}(\vec{u})){:}\Box B}$$

By the construction of $*$

$$\Delta^* \Rightarrow f_1{:}A_1^*,\ \ldots,\ \Delta^* \Rightarrow f_k{:}A_k^*, \quad y_1{:}A_1^*, \ldots, y_k{:}A_k^* \Rightarrow B^*(\vec{y})$$

for some proof polynomials $f_1, \ldots, f_k$. Also by the construction of $*$ from 6.2 we may assume, that the variables $y_1, \ldots, y_k$ do not occur in $A_1^*, \ldots, A_k^*$. By Lifting, find a proof polynomial $p(\vec{y})$ such that

$$y_1{:}A_1^*,\ \ldots,\ y_k{:}A_k^* \Rightarrow p(\vec{y}){:}B^*(\vec{y}).$$

By the substitution $[\vec{y}/\vec{f}]$ from the latter derivation we get

$$f_1{:}A_1^*,\ \ldots,\ f_k{:}A_k^* \Rightarrow p(\vec{f}){:}B^*(\vec{f}).$$

And by the induction hypothesis,

$$\vec{u}{:}\Delta^* \Rightarrow s_1^r(\vec{u}){:}f_1{:}A_1^*, \quad \ldots, \quad \vec{u}{:}\Delta^* \Rightarrow s_k^r(\vec{u}){:}f_k{:}A_k^*.$$

By Lifting (corollary 8.2),, construct a proof polynomial $g(\vec{x})$ such that

$$x_1{:}f_1{:}A_1^*,\ldots,x_k{:}f_k{:}A_k^* \Rightarrow g(\vec{x}){:}p(\vec{f}){:}B^*(\vec{f}),$$

by substitution,

$$s_1^r{:}f_1{:}A_1^*,\ldots,s_k^r{:}f_k{:}A_k^* \Rightarrow g(\vec{s^r}(\vec{u})){:}p(\vec{f}){:}B^*(\vec{f}),$$

and, by the transitivity of $\Rightarrow$,

$$\vec{u}{:}\Delta^* \Rightarrow g(\vec{s^r}(\vec{u})){:}p(\vec{f}){:}B^*(\vec{f}).$$

Let $(\mathbf{box}(t))^r = g(\vec{s^r}(\vec{u}))$.

At a $(\Box E)$-node of $\mathcal{D}$ we have a figure

$$\frac{\vec{u}{:}\Gamma \Rightarrow t(\vec{u}){:}\Box A}{\vec{u}{:}\Gamma \Rightarrow \mathbf{unbox}(t)(\vec{u}){:}A},$$

which corresponds to a natural deduction step from $\Gamma \Rightarrow \Box A$ to $\Gamma \Rightarrow A$. By the realization $*$ we have $\Gamma^* \Rightarrow A^*$. Use the standard transformation $(:E)$ from 9.5. to construct $h(\vec{x})$ and a pure proof

$$\vec{u}{:}\Gamma^* \Rightarrow h(\vec{u}){:}A^*.$$

Put $(\mathbf{unbox}(t))^r = h$.

◀

10 Conclusions

In this paper we concentrated on internal semantical and structural questions of $\mathcal{LP}$ and on building a decent realization of modal logic and λ-calculi in $\mathcal{LP}$. We have not addressed yet such an appealing issue as rewriting the entire $\mathcal{LP}$ in a pure λ way without proof constants but with extra operations on terms. In a way such a presentation of $\mathcal{LP}$ would correspond to a transition from the a combinatory system to a λ-style system. From the logical point of view this would mean a transliteration of proof terms from the Hilbert style basis to the natural deduction basis.

The Logic of Proofs has has a solid provability semantics and a more expressive language than either modal logic or the λ-calculus. The ability of the Logic of Proofs to emulate both modal logic and λ-calculus comes at some price. Modal logic and traditional λ-calculi cover only fractions of $\mathcal{LP}$ but instead enjoy nice symmetries, transparent models, normal forms, etc. Modal and λ presentations of the same facts are usually more compact than the corresponding presentations via proof polynomials. Modality and λ's may be regarded as higher level languages for corresponding fragments of Logic of Proofs.

Proof polynomials reveal the dynamic character of modality. The realization of $\mathcal{S}4$ in $\mathcal{LP}$ provides a fresh look at modal logic and its applications in general. Such areas as modal λ-calculi, polymorphic second order λ-calculi, λ-calculi with types depending on terms, nondeterministic λ-calculi, etc., could gain from their semantics as proof polynomials delivered by $\mathcal{LP}$.

Logic of Proofs for the second order arithmetic naturally requires quantifiers over propositions. Such a logic should accomodate second order polymorphic λ-calculi (cf. [15], [32], and supply them with provability semantics.

11 Acknowledgements

This work has benefited from many interactions over the past several years with a number of mathematicians, logicians and computer scientists: H. Barendregt, L. Beklemishev, J. van Benthem, G. Boolos, S. Buss, R. Constable, D. van Dalen, J. Diller, S. Feferman, E. Engeler, J.-Y. Girard, G. Jäger, D. de Jongh, V. Krupski, S. MacLane, G. Mints, F. Montagna, A. Nerode, E. Nogina, R. Parikh, W. Pholers, V. Pratt, A. Preller, J. Remmel, D. Roorda, A. Scedrov, R. Shore, T. Sidon, T. Strassen, A. Troelstra, A. Visser.

I am indebted to Volodya Krupski, Tanya Sidon and Fred Smith for a careful reading of this paper which led to valuable improvements.

References

[1] S. Artemov and T. Strassen, "The Basic Logic of Proofs", *Lecture Notes in Computer Science* , v. 702 (1993), pp. 14-28.

[2] S. Artemov and T. Strassen, "Functionality in the Basic Logic of Proofs", *Tech.Rep. IAM 92-004, Department for Somputer Science*, University of Bern, Switzerland, 1993.

[3] S. Artemov, "Logic of Proofs", *Annals of Pure and Applied Logic*, v. 67 (1994), pp. 29-59.

[4] S. Artemov, "Operational Modal Logic," *Tech. Rep. MSI 95-29*, Cornell University, December 1995.

[5] S. Artemov, "Proof realizations of typed λ-calculi," *Tech. Rep. MSI 97-2*, Cornell University, May 1997.

[6] J. van Benthem. "Reflections on epistemic logic", *Loguque & Analyse*, 133-134, pp. 5-14, 1991

[7] G. Bierman and V. de Paiva, "Intuitionistic necessity revisited", *Proceedings of the Logic at Work Conference,* Amsterdam (December 1992), Second revision, June 1996 (http://theory.doc.ic.ac.uk/tfm/papers.html).

[8] G. Boolos, *The Unprovability of Consistency: An Essay in Modal Logic*, Cambridge University Press, 1979

[9] G. Boolos, *The Logic of Provability*, Cambridge University Press, 1979

[10] V. A. J. Borghuis, *Coming to Terms with Modal Logic: On the interpretation of modalities in typed λ-calculus*, Ph.D. Thesis, Technische Universiteit Eindhoven, 1994

[11] S. Buss, "The Modal Logic of Pure Provability", *Notre Dame Journal of Formal Logic*, v. 31, No. 2, 1990

[12] A. Chagrov and M. Zakharyaschev, *Modal Logic*, Oxford Science Publications, 1997.

[13] D. van Dalen, *Logic and Structure*, Springer-Verlag, 1994.

[14] D. M. Gabbay, *Labelled Deductive Systems*, Oxford University Press, 1994.

[15] J.-Y. Girard, Y. Lafont, P. Tyalor, *Proofs and Types*, Cambridge University Press, 1989.

[16] K. Gödel, "Eine Interpretation des intuitionistischen Aussagenkalkuls", *Ergebnisse Math. Colloq.*, Bd. 4 (1933), S. 39-40.

[17] K. Gödel, "Vortrag bei Zilsel" (1938), in S. Feferman, ed. *Kurt Gödel Collected Works. Volume III*, Oxford University Press, 1995

[18] R. Goldblatt, "Arithmetical necessity, provability and intuitionistic logic", *Theoria*, 44, pp. 38-46, 1978.

[19] S. Kleene. "On the interpretation of intuitionistic number theory", *Journal of Symbolic Logic*, v. 10, pp. 109-124, 1945

[20] A. Kolmogoroff, "Zur Deutung der intuitionistischen Logik", *Math. Ztschr.*, Bd. 35 (1932), S.58-65.

[21] D. Kozen and J. Tiuryn, "Logic of Programs", in *Handbook of Theoretical Computer Science. Volume B, Formal Models and Semantics*, The MIT Press/Elsevier, pp. 789-840, 1990

[22] S. Kripke, "Semantical considerations on modal logic", *Acta Philosophica Fennica*, 16, pp. 83-94, 1963.

[23] V.N. Krupski, "Operational Logic of Proofs with Functionality Condition on Proof Predicate", Lecture Notes in Computer Science, v. 1234, *Logical Foundations of Computer Science' 97, Yaroslavl'*, pp. 167-177, 1997

[24] A.V. Kuznetsov and A.Yu. Muravitsky, "The logic of provability", Abstracts of the *4-th All-Union Conference on Mathematical Logic*, p. 73, (Russian), 1976.

[25] E. Lemmon, "New Foundations for Lewis's modal systems", *Journal of Symbolic Logic*, 22, pp. 176-186, 1957.

[26] S. Martini and A. Masini,"A computational interpretation of modal proofs", in Wansing, ed., *Proof Theory of Modal Logics*, (Workshop proceedings), Kluwer, 1994.

[27] P. Martin-Löf. "Constructive mathematics and computer programming", in *Logic, Methodology and Philosophy of Science VI*, North-Holland, pp. 153-175, 1982.

[28] P. Martin-Löf. *Intuitionistic Type Theory*, Studies in Proof Theory, Bibliopolis, Naples, 1984.

[29] Yu. Medvedev, "Finite problems", *Soviet Mathematics Doklady*, v. 3. pp. 227-230, 1962.

[30] E. Mendelson, *Introduction to mathematical logic. Third edition.*, Wadsworth, 1987

[31] G. Mints. "Lewis' systems and system T (1965-1973)". In *Selected papers in proof theory*, Bibliopolis, Napoli, 1992.

[32] J. Mitchell. "Type systems for programming languages". In van Leeuwen, ed. *Handbook of Theoretical Computer Science*, Elsevier/MIT press, 1990.

[33] A. Mkrtychev, "Models for the Logic of Proofs ", Lecture Notes in Computer Science, v. 1234, *Logical Foundations of Computer Science' 97, Yaroslavl'*, pp. 266-275, 1997

[34] R. Montague. "Syntactical treatments of modality with corollaries on reflection principles and finite axiomatizability", *Acta Philosophica Fennica*, 16, pp. 153-168, 1963.

[35] J. Myhill, "Some Remarks on the Notion of Proof", *Journal of Philosophy*, 57, pp. 461-471, 1960

[36] J. Myhill, "Intensional Set Theory", In: S. Shapiro, ed., *Intensional Mathematics*, North-Holland, pp. 47-61, 1985.

[37] C. Parsons and W. Sieg. "Introductory note to *1938*a*". In: S. Feferman, ed. *Kurt Gödel Collected Works. Volume III*, Oxford University Press, pp. 62-85, 1995.

[38] F. Pfenning and H.C. Wong, "On a modal lambda-calculus for S4", *Electronic Notes in Computer Science* 1, 1995.

[39] S. Shapiro. "Intensional Mathematics and Constructive Mathematics". In: S. Shapiro, ed., *Intensional Mathematics*, North-Holland, pp. 1-10, 1985.

[40] S. Shapiro. "Epistemic and Intuitionistic Arithmetic". In: S. Shapiro, ed., "Intensional mathematics", North-Holland, pp. 11-46, 1985.

[41] T. Sidon, "Provability Logic with Operations on Proofs", Lecture Notes in Computer Science, v. 1234, *Logical Foundations of Computer Science' 97, Yaroslavl'*, pp. 342-353, 1997

[42] R. Summyan, *Diagonalization and Self-Reference*, Oxford University Press, 1994

[43] G. Takeuti, *Proof Theory*, North-Holland, 1975

[44] A.S. Troelstra and D. van Dalen, *Constructivism in Mathematics. An Introduction*, v. 1, Amsterdam; North Holland, 1988.

[45] A.S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Cambridge University Press, 1996.